

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA

Grado en Ingeniería Informática

Desarrollo de plataforma para la gestión de los TFG

Development of a final degree projects management platform

Realizado por

Juan José Ariza Fernández

Tutorizado por

Eduardo Guzmán de los Riscos

Departamento

Lenguaje y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

MÁLAGA, octubre, 2016

Fecha defensa:

El Secretario del Tribunal

Resumen

La realización de este trabajo fin de grado pretende, por un lado plasmar los conocimientos adquiridos en los cuatro años de estudios en Sistemas de la Información, así como en lo posible, tratar de explorar y añadir nuevos conocimientos en materia de desarrollo de aplicaciones Web. Para ello usaremos tecnologías como PHP, MySQL, Javascript, JQuery, Ajax, HTML5 y CSS3.

También, siguiendo el auge de los dispositivos móviles como vehículo de acceso a Internet, se diseñará un portal Web, que buscará la facilidad de uso y efectos visuales llamativos, adaptándose a las características de dichos dispositivos.

Como patrón de diseño para implementar el Sistema se utilizará el conocido Modelo Vista Controlador (MVC), arquitectura estándar para aplicaciones interactivas y que separa los datos y la lógica de negocio de una aplicación de las vistas del usuario. Este patrón de diseño facilita el desarrollo y posterior mantenimiento de las aplicaciones gracias a la reutilización de código y a la separación de conceptos.

Por otro lado, se quiere construir una aplicación Web cuya funcionalidad ayude a todo el personal relacionado con los TFG. Facilitando así, el desarrollo por parte del alumnado, la tutorización detallada del profesor tutor, el aporte de información al tribunal que sea hará cargo de la lectura para una mayor comprensión del desarrollo, y una mayor fluidez en la administración para todo lo relacionado con la petición de presentar el TFG.

Palabras clave

- Plataforma web
- Multiplataforma
- Gestión
- Seguimiento
- Trabajo fin de grado
- Convocatoria
- Lectura
- Tribunal
- Administración
- Alumno
- Profesor

Summary

The aims of the realization of this final degree Project (FDP) are capturing the knowledge we have acquired in these four years of studying the Information Systems field and trying to explore and add new knowledges about web development. For this, we will use technologies like PHP, MySQL, JavaScript, JQuery, Ajax, HTML5 and CSS3.

Also, following the rise of mobile devices like internet access, we will design a web platform which will look for ease of use and eye-catching visual effects, adapting to the characteristics of these devices.

We will use the Model-View-Controller (MVC) design pattern to implement the system. This architecture is standard for interactive applications and it separates the data and the business logic of an application from the user views. This design pattern facilitates the development and subsequent maintenance of applications by reusing code and separating concepts.

On the other hand, it has been proposed to develop a web application whose functionality is to help all personal involved in the FDP. Thus facilitating the development by the student, the review by the teacher, the provision of information to the court for a better understanding of the development, and finally, greater fluidity in the administration for everything related to the request to present the FDP.

Keywords

- Web platform
- Multi-platform
- Management
- Monitoring and control
- Final Degree Project
- Announcement
- Reading
- Court
- Administration
- Student
- Teacher

Índice

1. Introducción	7
Objetivos	7
Métodos	7
Fases de trabajo	8
2. Procesos de negocio.....	9
Rol Usuario	9
Rol Alumno	9
Rol Profesor Tribunal.....	11
Rol Administrador	12
3. Descripción general del entorno tecnológico	14
Metodología de análisis y diseño	14
Utilización de tecnologías basadas en el concepto web 2.0	14
PHP	14
Modelo de capas	15
Utilización del patrón MVC	15
Front controller	17
Framework Bootstrap	17
Control de acceso.....	17
Base de datos	18
Patrón Singleton.....	18
Servidor Web.....	18
Entorno de desarrollo	19
4. Análisis del sistema	20
Descripción del funcionamiento del sistema	20
Catálogo de requisitos.....	21
Casos de Uso	25
5. Diseño del Sistema	61
Diseño Lógico de Datos	61
Diseño Físico de Datos	64
6. Implementación	65
Detalle de las tecnologías.....	65
Preparación del proyecto.....	66

Detalles de Estructuración	72
Implementación de una funcionalidad	75
7. Casos de Prueba	80
9. Conclusiones y Trabajo Futuro	87
Anexo I - Manual de Usuario.....	88
Bibliografía	103

1. Introducción

Objetivos

Este proyecto surge ante la necesidad de disponer en la E.T.S.I. Informática de una herramienta que permita la gestión eficiente de todos los procesos relacionados con los Trabajos de Fin de Grado (TFG), ya que el proceso actual para esta tarea se hace mediante otros métodos como entrega en mano, correo electrónico, etc.

El objetivo de este proyecto es la realización de una herramienta enfocada a todos los ámbitos de los TFG, en el ámbito administrativo, del alumnado, y de los tutores.

El enfoque que se pretende seguir se puede resumir de la siguiente forma:

- Para el alumnado, se pretende crear una herramienta en la que puedan subir todos los productos generados como parte del TFG (código fuente, memoria, etc.), para que puedan ser revisados, creando así una línea de trabajo donde quedarán registrados todos los procesos que se siguen, así como las correcciones de los tutores.
- Para el profesorado tutor de los TFG, se pretende que mediante esta herramienta puedan llevar un control más detallado y más cómodo de los TFG que se van a presentar a la convocatoria correspondiente de lecturas.
- Para el profesorado miembro de los tribunales de TFG, que podrán acceder a la información y documentación del alumnado asignado a la sesión de lectura de los que estos profesores son miembros del tribunal.
- Para la parte administrativa, se pretende crear esta herramienta para facilitar los trámites que conlleva realizar el TFG, tanto en entrega de documentos por parte del alumno, como trámites correspondientes a la lectura y defensa del TFG.

La herramienta se encargará de que los trámites para la realización de un TFG se hagan de una forma adecuada y con un seguimiento detallado para que, en caso de cualquier problema, pueda revisarse de forma correcta y con la mayor brevedad posible.

Métodos

Para lograr los objetivos anteriores, se va a adaptar una herramienta open source para la gestión de las conferencias denominada *Conference Systems*. De esta aplicación obtendremos la estructuración básica donde montar nuestra plataforma.

Una vez obtenido el esquema principal de la plataforma, comenzaremos con la adaptación de su modelo de base de datos, para adaptarlo a las necesidades de este TFG.

Posteriormente empezaremos diseñando las funcionalidades básicas que controlen la plataforma (*back-end*). Las conexiones con el servidor y la base de datos, el controlador de cada vista de la plataforma, como el proceso de autenticación encargado de acceder a una sección diferente dependiendo del rol, o una vez autenticado la vista específica que tendrá cada tipo de rol y las funcionalidades a las que tendrán acceso.

Una vez estructurada la parte de *back-end*, se comienza la realización del *front-end*, es decir, la parte visual de la plataforma. Lo primero será crear una plantilla general para toda la plataforma, e ir añadiendo diferentes elementos a esta plantilla dependiendo de la funcionalidad de esa vista.

Una vez creada la parte de *front-end* y *back-end*, se harán las conexiones pertinentes entre estos junto con la base de datos, perfilando y modificando los elementos que no queden correctamente.

Por último, pasaremos una fase de pruebas a la plataforma para que todo quede debidamente comprobado y que cumple la funcionalidad definida, dando así una versión completamente funcional.

Fases de trabajo

Para el desarrollo del proyecto se aplicará una metodología iterativa incremental estructurada en iteraciones. Previamente se desarrollarán las siguientes tareas:

1. Estudio de la aplicación web Open Conference Systems.
2. Diseño E/R de la base de datos
3. Diseño general de la arquitectura de la aplicación.

Posteriormente, dentro de cada ciclo de iteración se realizarán las siguientes fases:

Revisión de la aplicación web Open Conference Systems.

1. Análisis y especificación
2. Diseño
3. Implementación del *back-end*
4. Implementación del *front-end*
5. Pruebas
6. Documentación

2. Procesos de negocio

Las tareas que se desean implementar se dividirán en los siguientes procesos de negocio en relación con los diferentes roles que los usuarios pueden adoptar.

Rol Usuario

Acceso a la plataforma web

El acceso a la plataforma web por parte de un usuario se hará con su DNI y una contraseña, la cual, en la creación de cada usuario se designará con su DNI, que podrá ser cambiada posteriormente.

Una vez iniciada la sesión en la plataforma web se redirigirá al usuario dependiendo del rol que adopten.

Identificación en la web

Cada usuario tendrá un panel donde podrá ver su información personal, así como modificarla en caso de que fuera necesario.

Panel de notificaciones

Cada usuario tendrá un panel de notificaciones, donde se le indicará cada actividad nueva que no ha sido aún revisada, redirigiéndolo al recurso que representa a dicha actividad.

Rol Alumno

Entrega anteproyecto

Cada alumno tendrá acceso a un formulario por el cual hará entrega de su anteproyecto, además de sus datos, el de su tutor y el del proyecto que va a realizar. Esta solicitud llegará a la administración y ellos se harán cargo de procesar la petición y abrir una línea de trabajo para ese proyecto.

Realizar entregas

Cada alumno tendrá un panel dirigido para las entregas que realiza en su TFG. Estas entregas quedaran contempladas con fecha y hora. Cada entrega poseerá un panel de detalle de entrega, donde se podrá acceder al título de la entrega, la descripción y los archivos adjuntados. Además de esta información se añadirá un panel de comentarios para cada entrega donde el alumno podrá conversar con su tutor/es acerca de la misma.

Añadir que todos los comentarios realizados por parte de el/los alumno/s quedarán reflejados y ordenados con fecha y hora, además de diferenciados dependiendo del usuario que los escriba.

Petición de convocatoria

Cuando el alumno quiera presentar su proyecto, hará una petición a su tutor, el cual aceptará el proyecto cuando crea conveniente. Una vez aceptada esta petición, el alumno podrá seleccionar de las convocatorias abiertas a la cual presentarse. Una vez la administración le asigne una lectura, éste tendrá acceso a ella.

Acceso a la lectura

Cuando sea asignado el día de la lectura del TFG de un alumno, éste será notificado, tanto en el panel de notificaciones, como en el panel de entregas del proyecto, pudiendo acceder a toda la información sobre la lectura (Fecha, Aula, Hora y Tribunal asignado).

Entrega de presentación

Una vez abierta la lectura, el alumno tendrá acceso a una sección donde realizar una entrega final, junto con la memoria y la presentación para la lectura.

Detalles de la lectura

Una vez que el alumno realice su presentación ante los tribunales, en la sección de detalles de la lectura se abrirá una vista de detalles donde tanto su tutor como el tribunal indicarán su valoración junto con su nota del trabajo.

Rol Profesor Tribunal

Tutorización

Cada profesor tendrá un panel de tutor, donde se le mostrarán los proyectos que tiene asignados, así como las entregas que se realizan en cada uno de ellos. Como información adicional, también se le notificará al profesor cuando una lectura sea asignada para uno de los proyectos del que es tutor, pudiendo acceder a toda la información de dicha lectura.

Seguimiento de los proyectos

Cada proyecto asignado a un profesor tendrá un panel de detalle, donde se especificarán tanto el título del proyecto, su descripción, el/los tutores asignado y el/los alumnos asignados. Además de un panel donde se especificará cada entrega que se ha realizado por parte de los alumnos.

Estas entregas también poseerán un panel de detalle de entregas, similar al de los alumnos, donde el tutor podrá ver los detalles de la entrega (título, descripción, archivos adjuntos), y podrá realizar comentarios a dicha entrega, para que el alumno pueda tener una idea de cómo progresa su proyecto.

Añadir que todos los comentarios realizados por parte del tutor quedaran reflejados y ordenados con fecha y hora, además de diferenciados dependiendo del usuario que los escriba.

Evaluación de los alumnos

Cada tutor tendrá acceso a la lectura asignada a sus proyectos, teniendo una sección en la que podrá añadir una valoración del trabajo que ha realizado el alumno, y una nota. Esta información será proporcionada al tribunal para su propia evaluación.

Tribunal

Cada profesor, además de ser tutor, podrá ser tribunal, por tanto, tendrá un panel donde poder ver cada lectura a la que asistirá como tribunal, así como la información de dicha lectura (Proyecto, día, fecha, hora y tribunal asignado).

Gestión de lecturas

Cada profesor que haya sido nombrado como miembro de tribunal tendrá acceso a un panel donde aparecerán las lecturas que se realizarán en su convocatoria. Podrá asimismo manifestar sus preferencias en cuanto a los TFG más afines a su conocimiento y, por tanto, de

los que preferiría ser miembro del tribunal. Una vez el administrador haya asignado al tribunal, cada miembro será avisado.

Evaluación de presentación

El tribunal asignado a la lectura de cada TFG tendrá acceso a la entrega final de presentación de cada alumno y la evaluación realizada por el tutor. Además, podrá visualizar una sección donde el presidente podrá rellenar la evaluación del tribunal y asignar una nota al TFG junto con una valoración.

Rol Administrador

Gestión

Cada administrador tendrá acceso a la gestión de usuarios, proyectos, solicitudes y convocatorias. Pudiendo tanto crear, borrar, obtener o actualizar cada uno de los apartados anteriores.

Gestión usuarios

El administrador tendrá acceso a los usuarios que se encuentra en el sistema diferenciados por su rol. Además de poder hacer un CRUD de usuarios en el sistema.

Gestión proyectos

El administrador tendrá acceso al detalle de cada proyecto abierto que se encuentre en el sistema. Además de poder hacer un CRUD de proyectos en el sistema.

Gestión convocatorias

El administrador tendrá acceso a las convocatorias que hay abiertas en el sistema, pudiendo acceder al detalle de cada una, donde encontrará un desglose de los días de cada convocatoria, y las lecturas asignadas a ese día. Además, tendrá acceso al detalle de cada lectura, pudiendo asignar o cambiar tribunales para cada una de ella. Añadir que tendrá acceso tanto al CRUD de convocatorias, lecturas y tribunales.

Gestión solicitudes

El administrador tendrá acceso a las solicitudes que se hagan para la creación de un nuevo TFG. En este apartado el administrador tendrá la opción de generar los alumnos automáticamente, y además generar el proyecto y asignar a los tutores que sean necesarios.

3. Descripción general del entorno tecnológico

Metodología de análisis y diseño

Para la especificación del problema y el diseño de la solución, emplearemos UML (Unified Modelling Language) en su versión 2.X.

Utilización de tecnologías basadas en el concepto web 2.0

Para la implementación de la aplicación se usarán determinadas tecnologías que se sitúan dentro del esquema de la Web2.0, esquema que marca una tendencia hacia la facilidad de interacción del usuario con la interface del lado del cliente (front-end).

- Dar funcionalidad y aspecto de la plataforma Web como si fuera software de escritorio
- Respeto a los estándares W3C
- Separación del contenido del diseño con uso de hojas de estilo CSS3
- HTML5 para la definición de contenido de la Web
- Utilización de JavaScript, JQuery y Bootstrap en el front-end para mejorar el aspecto de las interfaces del cliente
- Ajax junto con JSON para el intercambio de información entre el servidor y el cliente

PHP

En el lado del servidor se usará PHP Orientado a Objetos, ya que se trata de una tecnología de Open Source de las que existen gran cantidad de herramientas y Frameworks gratuitos, existiendo también multitud de documentación y APIs de desarrollo, siendo especialmente útil para agilizar el desarrollo de la práctica.

Otros motivos por los que escoger PHP:

- Partir de otra aplicación con funcionalidad similar, como es Open Conference Systems.
- Curva de aprendizaje muy baja
- Los entornos de desarrollo son de rápida y fácil configuración
- Fácil despliegue
- Fácil acceso a base de datos
- Comunidad muy grande
- Multiplataforma
- SFS (Estabilidad, flexibilidad y velocidad)
- Interoperable con otras tecnologías: JavaScript, HTML, JSON, etc.

Modelo de capas

Se ha elegido, dentro de la arquitectura cliente/servidor, un modelo de tres capas. El modelo de capas es una técnica software para separar las diferentes partes de la aplicación, con el objetivo de mejorar su rendimiento, mantenimiento y sus funciones. Esta separación de las diferentes partes hace que los cambios en cualquiera de las capas no afecten o afecten poco a las otras capas en que está dividida la aplicación. Las tres capas serían las siguientes:

Capa de presentación: Es la capa que ve el usuario, presenta el sistema, captura la información y la presenta la información al usuario en un mismo proceso. Esta capa se comunica únicamente con la capa de negocio.

Capa de negocio o lógica: Se reciben las peticiones del usuario y se envían las respuestas tras el proceso. En esta capa se establecen todas las reglas que deben cumplirse. Se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, también se comunica con la capa de persistencia o de datos para solicitar al gestor de la base de datos para recuperar, modificar o insertar datos en la base de datos.

Capa de persistencia o de datos: Es donde residen los datos y la encargada de acceder a ellos. Está formada por uno o más gestores de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación desde la capa de negocio.

Utilización del patrón MVC

Dentro del modelo de arquitectura en tres capas uno de los patrones de diseño de software más utilizado es el patrón Modelo-Vista-Controlador (MVC).

El patrón de diseño MVC fue descrito por primera vez en el año 1979 por Trygve Reenskaug, de los laboratorios de I+D de Xerox. Su característica primordial es la separación de la aplicación en tres capas distintas: el Modelo, la Vista y el Controlador.

Así, el MVC es una guía para el diseño de arquitecturas de aplicaciones que mantienen una gran interactividad con los usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información y el tercero es un conjunto de controladores que procesan las peticiones de los usuarios y controla el flujo de ejecución del sistema.

En el siguiente gráfico se observa el intercambio de información entre las diferentes capas del patrón MVC. El cliente realiza una petición al controlador, éste obtiene los datos del modelo, y los presenta al cliente mediante la vista.

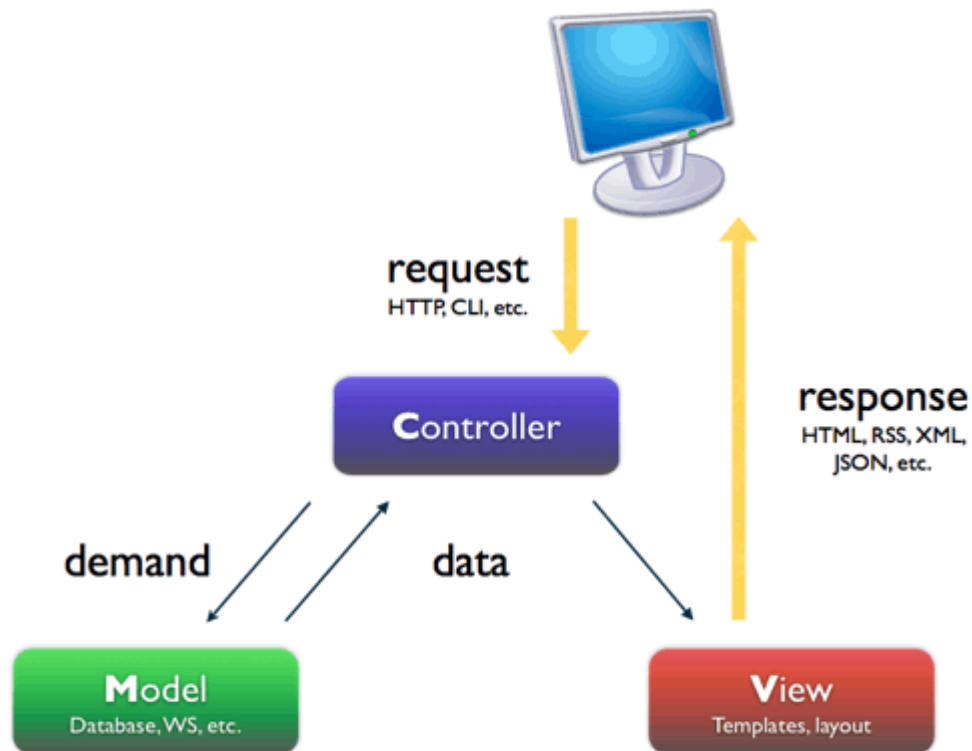


Figura 1. Modelo Vista Controlador.

Por tanto, el MVC divide la arquitectura de una aplicación en 3 tipos de clases fundamentales, con las responsabilidades siguientes:

Clase Modelo:

- Procesa peticiones del sistema
- Genera datos de respuesta del sistema
- Gestiona y almacena la información

Clase Vista:

- Genera peticiones
- Muestra respuestas del sistema

Clase Controlador:

- Gestiona procesamiento de peticiones
- Gestiona las respuestas del sistema

Por tanto, en este proyecto, identificamos a las 3 clases como:

- La **clase modelo** será representada por las clases de la base de datos en MySQL, y las clases que nos permite hacer PHP Orientado a Objetos.
- La **clase controlador**, y encargada de las peticiones, será representada por Ajax junto con PHP, y JSON para el intercambio de mensajes.
- La **clase vista** será representada por HTML5, JavaScript y CSS3, usando el Framework Bootstrap.

Front controller

Además del patrón MVC usaremos el controlador frontal (front controller), el cual implementaremos en cada vista, haciéndose cargo del control de acceso y de recuperar los datos necesarios para esa vista.

Framework Bootstrap

Del lado del cliente, Front-End, se utilizará el Framework Bootstrap.

Bootstrap es un framework que simplifica el proceso de creación de diseños web combinando CSS y JavaScript, desarrollado por Twitter.

Su mayor ventaja es que se pueden crear interfaces de usuario que se adaptan a los diversos navegadores actuales. Cuenta con multitud de componentes Web, lo que nos ahorra tiempo en el diseño y construcción de las interfaces del usuario.

Además, se integra perfectamente con las principales librerías JavaScript, como JQuery, utilizada también en la realización de este proyecto.

Ofrece un diseño sólido usando LESS (Lenguaje de Hojas de Estilo Dinámico) y estándares como CSS3/HTML5.

Funciona con todos los navegadores, incluido Internet Explorer usando HTML Shim para que reconozca los tags HTML5.

Control de acceso

Para el control de acceso a la plataforma Web, usaremos como anteriormente dijimos el lenguaje PHP combinado con el uso de sesiones. Estas sesiones serán creadas cuando un usuario acceda a la plataforma Web mediante su DNI y su contraseña. El controlador de cada vista será el encargado de comprobar que el usuario inició correctamente la sesión, impidiendo así que cualquier usuario acceda a la plataforma sin ser autenticado.

Una vez autenticado el usuario, una clase de redirección será encargada de facilitar el acceso a determinadas zonas dependiendo del rol que posea el usuario, diferenciando así las zonas de administración, tutorización y alumnado.

Base de datos

Para la realización de la capa de modelo, como base de datos hemos optado por usar MySQL, ya que es un SGBD relacional y libre, publicado bajo la licencia dual GPL/Licencia comercial.

Entre sus ventajas destacamos:

- Open Source
- Velocidad al realizar operaciones
- Bajo costo en requerimientos para la elaboración de las Bases de Datos, ya que debido a su bajo consumo puede ser ejecutado en una maquina con escasos recursos sin ningún problema
- Facilidad de configuración e instalación, además de ser multiplataforma
- Baja probabilidad de corromper datos
- Soporte de transacciones
- Escalabilidad, estabilidad y seguridad
- Soporta procedimientos almacenados
- Permite uso de comandos DDL y DML

Patrón Singleton

Para la conexión entre la base de datos y nuestras clases definidas en PHP usaremos el patrón singleton.

Con el uso de este patrón, también llamado de instancia única, se pretende restringir la creación de objetos pertenecientes a una clase a sólo uno. Garantizando que sólo hay una instancia y proporciona un único punto de acceso. Con esto no haremos un uso innecesario de recursos.

Servidor Web

Para la realización del proyecto se ha optado, por su sencillez y extenso uso, a Apache como servidor Web. Funciona como un servidor HTTP, es decir, se encarga de enviar peticiones desde la conexión del cliente hasta el sistema que se encarga de administrar cada tipo de petición, de este modo se transmite la respuesta correspondiente de nuevo al cliente.

Dicho servidor es mantenido en buena parte por la Apache Software Foundation y es de uso gratuito.

Algunas de las ventajas de usar Apache son:

- Modular
- Código abierto
- Multiplataforma
- Extensible
- Gran comunidad

Entorno de desarrollo

Para el desarrollo de la aplicación Web en primer lugar usaremos la herramienta XAMPP, la cual nos da acceso a un servidor Apache junto con MySQL y PHP, facilitando la instalación de estos mediante un auto instalador. Proporcionando además un panel de control para la base de datos como es phpMyAdmin.

Para la escrita del código usaremos Sublime Text 3, ya que es una potente herramienta que nos permite tener acceso a cualquier tipo de código, además de facilitarnos la programación de infinidad de lenguajes. Además, posee una gran comunidad por detrás, proporcionando diferentes plugins para mayor facilidad a la programación.

4. Análisis del sistema

En el análisis del sistema vamos a realizar un análisis teórico de la plataforma Web. Se hace una descripción general del funcionamiento del sistema y a continuación se detallan tanto sus requisitos funcionales como los no funcionales.

Descripción del funcionamiento del sistema

La plataforma Web pretende cubrir todas las necesidades de todas las partes implicadas en el desarrollo de un TFG. Tanto a los alumnos que lo desarrollan, los tutores que guían el desarrollo, los tribunales que los evalúan y la administración que gestiona el proceso.

Los usuarios con rol alumno, podrán hacer una petición de desarrollo del TFG a través de la plataforma, haciendo entrega del anteproyecto, rellenando cierta información personal y sobre el TFG. Estas peticiones serán procesadas por el personal de administración y se generarán los datos necesarios para que tanto el alumno como el tutor puedan acceder a los detalles del TFG.

La aplicación debe permitir a cada usuario dependiendo de su rol tener acceso a los proyectos en los que estén implicados. De esta forma, un usuario que acceda a la web puede seguir un proyecto de forma detallada y precisa, facilitando así tanto el desarrollo como la evaluación.

Además del acceso al proyecto y sus detalles, la plataforma ofrece al usuario la posibilidad de ver qué convocatorias están abiertas, la gestión de las lecturas, y la presentación de documentos para la lectura.

Una vez realizada la descripción del sistema se analizará más en detalle los requisitos de la plataforma.

Para la obtención de los requisitos del sistema, hemos tomado como referencia la aplicación web Open Conference System, información acerca de la realización de un TFG en la web de la Escuela, a través del tutor, explicando cuál es el proceso para la realización del TFG, y a través de la propia experiencia.

Existen dos tipos de requisitos:

- Funcionales: Describen el comportamiento del Sistema, qué acciones ha de desarrollar y que resultados debe de ofrecer.
- No Funcionales: Especifican características que ha de cumplir el Sistema, en términos de qué datos se deben almacenar, de calidad y de seguridad.

Catálogo de requisitos

Identificación de Usuarios

Se han identificado cuatro clases de Usuarios del Sistema:

Alumno

El Usuario alumno es el que realiza las peticiones para la realización de un TFG. A través de esta petición comienza el funcionamiento del sistema, ya que es esta petición la que genera el proyecto, pone en contacto al tutor, y genera el usuario en el sistema correspondiente al alumno. A través de este usuario podrá acceder a la plataforma y realizar entregas de su TFG, así como comentarlas con su tutor.

Otra funcionalidad para el alumno es la solicitud de la lectura del TFG para una determinada convocatoria. Una vez sea aceptada por su tutor, el alumno podrá elegir la convocatoria correspondiente. Una vez seleccionada la convocatoria, se le abrirá una sección a la que subir una entrega final junto con la memoria de su TFG para la presentación. Por último, en esta misma sección podrá ver la valoración tanto de su tutor, como la del tribunal, cuando esta se publique.

Profesor

El Usuario profesor es el que se encarga de ver en detalle el proyecto del alumno, teniendo acceso a cada entrega que realiza y pudiendo hacer evaluaciones de ésta.

También es el encargado de aceptar el proyecto para que el alumno pueda hacer su presentación. Además de tener una sección de evaluación del alumno para referencia del tribunal.

Tribunal

El Usuario tribunal, además de ser un profesor, es el encargado de valorar la defensa de los TFG. Cada tribunal asignado a una lectura tendrá un Presidente, el cual, tendrá una sección para ver qué entregó el alumno, la evaluación de su tutor, y una sección donde evaluar y calificar al alumno.

Administración

El Usuario administrador, es el encargado de gestionar todo el proceso que conlleva hacer un TFG. Desde aceptar el Anteproyecto, generar el proyecto y los usuarios, asignar los proyectos a una lectura, y tener información sobre la entrega final del alumno.

Requisitos funcionales (Comunes)

RF01. Identificación de usuario. Cada usuario tendrá acceso a la plataforma mediante su DNI y una contraseña. En el primer acceso, su contraseña equivaldrá a su DNI. Una vez en la plataforma cada usuario poseerá un panel de identificación, junto con su rol o el proyecto en el que estén participando.

RF02. Notificaciones. Cada usuario poseerá un panel de notificaciones, donde recibirán un mensaje en caso de algún cambio pertinente en el que esté involucrado.

RF03. Detalles de un proyecto. Cada proyecto que se encuentre en el sistema tendrá una vista pública, con datos principales, para cualquier usuario que este identificado en el sistema.

RF04. Perfil de usuario. Cada usuario tendrá acceso a su perfil, pudiendo modificar cada dato (Foto, nombre, contraseña, ...), a excepción de su DNI.

Requisitos funcionales (Alumno)

RF05. Solicitar TFG. Cada alumno podrá acceder a un apartado de la plataforma donde realizar una petición para realizar su TFG. En este apartado deberá rellenar sus datos personales, los datos de su TFG, realizar la entrega del anteproyecto y la solicitud para realizar el TFG, y por último indicar una contraseña, que se podrá solicitar en Secretaría, para que su solicitud sea válida. Una vez haya sido revisado por la administración se le generará el usuario y el proyecto.

RF06. Gestión de Entregas. Cada alumno, al ser asignado a su proyecto, podrá acceder a un panel de gestión de entregas, en el que podrá crear, ver y eliminar la entrega. Cada una de las entregas poseerá un panel de comentarios, en el que tanto el alumno como el profesor podrán interactuar entre sí para opinar sobre dicha entrega.

RF07. Petición de convocatoria. Una vez se hayan realizado varias entregas (al menos una), se le habilitará un botón al alumno para pedir a su tutor aceptación para presentarse a convocatoria. Una vez el tutor acepte esta petición, al alumno se le habilitará otro botón con el que poder seleccionar una de las convocatorias abiertas. Cuando la administración se haga cargo de esta petición y asigne una lectura al proyecto, se notificará al alumno con los detalles de la misma.

RF08. Entrega final. Una vez se le haya asignado la lectura al alumno, éste podrá realizar una entrega final (memoria, proyecto, ...), la cual podrá ser revisada tanto por el tutor, como por el tribunal asignado a la lectura.

RF09. Revisión de la presentación. Una vez el alumno haya realizado su presentación y su entrega final, podrá tener una revisión de la valoración y la nota asignada por su tutor y por el tribunal de la lectura. Con estas dos notas y mediante la ponderación de la escuela, se le asignará al alumno una nota final a su TFG.

Requisitos funcionales (Profesor)

RF10. Proyectos asignados. Cada profesor tendrá un panel donde aparecerán los proyectos de los que sea tutor. En él aparecerán el nombre del proyecto, las entregas que el alumno ha realizado, y un panel de aviso donde indicará el estado en el que se encuentra el proyecto, y si el proyecto está pendiente de autorización, pendiente de asignación de lectura, o con lectura asignada.

RF11. Detalles del proyecto. Cada proyecto tendrá un panel de detalle donde el profesor tendrá acceso a cada dato del proyecto y a las entregas realizadas por el alumno, junto con su fecha y hora de entrega.

RF12. Detalles de la entrega. Por cada entrega, al igual que en el caso del alumno, tendrá un panel de detalles, donde se indicará el autor de la entrega, el título, una descripción y los archivos adjuntos que el alumno haya realizado. En este panel el profesor podrá intercambiar comentarios con el autor de la entrega (alumno).

RF13. Autorización de proyectos. Cada tutor tendrá la responsabilidad de aceptar o no la petición del alumno de presentar su convocatoria.

RF14. Valoración del alumnado. Una vez asignada la lectura a un proyecto, se le notificará al tutor para que realice una valoración sobre el trabajo realizado el alumno, para posterior revisión por el tribunal de la lectura.

Requisitos funcionales (Tribunal)

RF15. Lecturas asignadas. Cada profesor tendrá un panel donde se muestren las lecturas a las que ha sido asignado. Mediante este panel podrá acceder a los detalles de cada lectura.

RF16. Gestión de lecturas. Cada profesor, tendrá un panel de pujas para las lecturas, donde se mostrarán todas las lecturas abiertas, y podrán expresar sus preferencias, y añadir un comentario adicional, si así lo desea.

RF17. Detalles de la lectura. Cada lectura que sea asignada a un profesor, poseerá un panel de detalle, donde se indicarán el día, hora, aula, tribunal asignado y detalles del proyecto.

RF18. Evaluación. Cada lectura tendrá asociado un tribunal, el cual tendrá acceso a la entrega final que realizó el alumno y a la valoración que realizó el tutor. Una vez revisados, el Presidente del tribunal junto con los otros miembros podrá evaluar al alumno mediante un panel en el detalle de la lectura, quedando reflejado en el mismo la entrega del alumno, la valoración del tutor y la valoración del tribunal.

Requisitos funcionales (Administrador)

RF19. **Gestión de solicitudes.** Cuando un alumno realiza una petición mediante una solicitud, esta es revisada por un personal de la administración, el cual se encarga de comprobar los documentos entregados, generar el usuario para el alumno, y generar el proyecto. El proceso de generación de usuario y proyecto será automático.

RF20. **CRUD Usuarios.** Cada administrador poseerá un panel de gestión de usuarios, en el que podrá crear, ver, modificar o borrar un usuario del sistema.

RF21. **CRUD Proyectos.** Cada administrador poseerá un panel de gestión de proyectos, en el que podrá crear, ver, modificar o borrar un proyecto del sistema. Además, posee un buscador de proyectos por categorías.

RF22. **CRUD Convocatorias.** Cada administrador poseerá un panel de gestión de convocatorias, donde podrá crear, ver, modificar o borrar una convocatoria.

RF23. **CRUD Lecturas.** Una vez creada una convocatoria, cada administrador poseerá un panel donde gestionar la lectura de cada proyecto, pudiendo especificar día, hora, lugar y tribunal.

Requisitos no funcionales

RNF01. El sistema deberá implementar la validación de la entrada de datos mediante JavaScript.

RNF02. El sistema deberá implementar tecnologías inscritas dentro del marco Web 2.0, JSON, JQuery, Ajax, CSS3, HTML5, etc.

RNF03. Como patrón de diseño para implementar el Sistema se utilizará el Modelo Vista Controlador (MVC).

RNF04. Como formato de intercambio de datos entre el modelo de datos alojado en el Servidor y el Cliente, se utilizará JSON.

RNF05. Solo los usuarios identificados podrán acceder a la plataforma. A su vez, cada rol de usuario diferenciará cada zona de la plataforma, dando acceso único a cada zona dependiendo del rol.

RNF06. Los recursos de un proyecto y todo lo que conlleva (entregas, lecturas, archivos, ...) solo será accesible a los usuarios involucrados en el (alumnos, tutores, ...).

RNF07. Las contraseñas de los usuarios serán encriptadas.

RNF08. Un proyecto solo podrá asignarse a una lectura si el tutor autorizó la petición del alumno, y éste eligió la convocatoria, además de que no haya sido asignado ya a una lectura. Estado del proyecto: "Lectura pendiente".

RNF09. Un profesor será asignado como tribunal a una lectura siempre que no pertenezca como tutor a ese proyecto.

RNF10. Solo se podrá asignar un presidente por lectura.

RNF11. El control de acceso a la plataforma se hará mediante las Sesiones de PHP.

RNF12. Las relaciones de la base de datos se harán cargo de actualizar o borrar cualquier recurso que tenga otros recursos relacionados con el (Clave Foránea con Actualizar o borrar en cascada).

Casos de Uso

El modelo de casos de uso especifica la funcionalidad que el sistema ha de ofrecer desde la perspectiva de los usuarios y lo que el sistema ha de realizar para satisfacer las peticiones de estos usuarios.

En función a los requisitos funcionales descritos anteriormente encontramos estos cuatro escenarios diferenciados por el tipo de usuario que accede a la plataforma.

Usuarios (Todos en común)

En el siguiente diagrama se muestran las funcionalidades que posee todos los usuarios de la web.

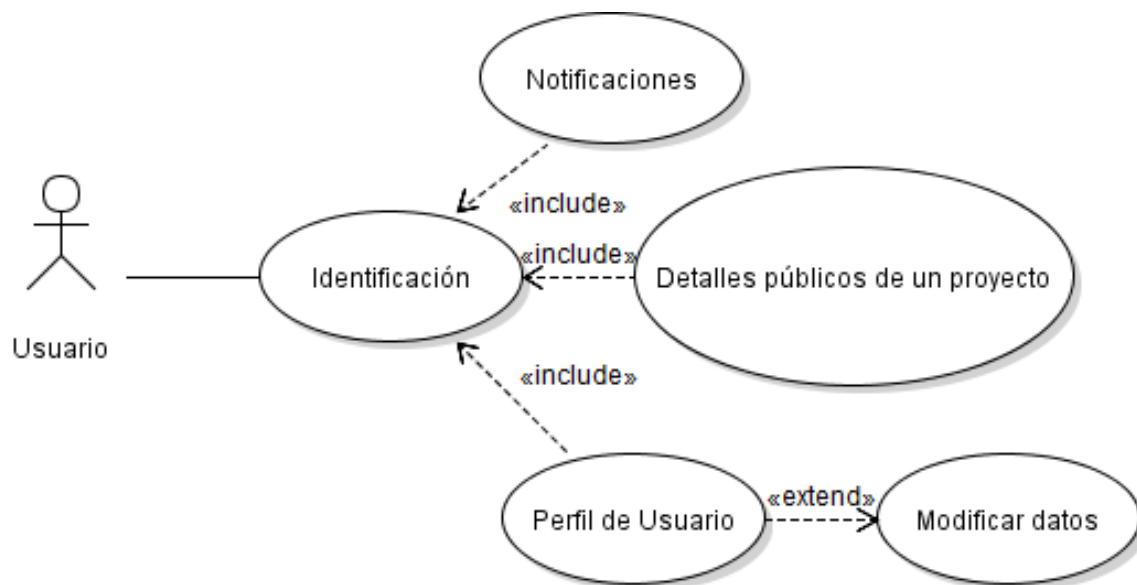


Figura 2. Diagrama de Casos de Uso – Rol Usuario.

Usuario Alumno

En el siguiente diagrama se muestran las funcionalidades que poseen los usuarios con rol alumno.

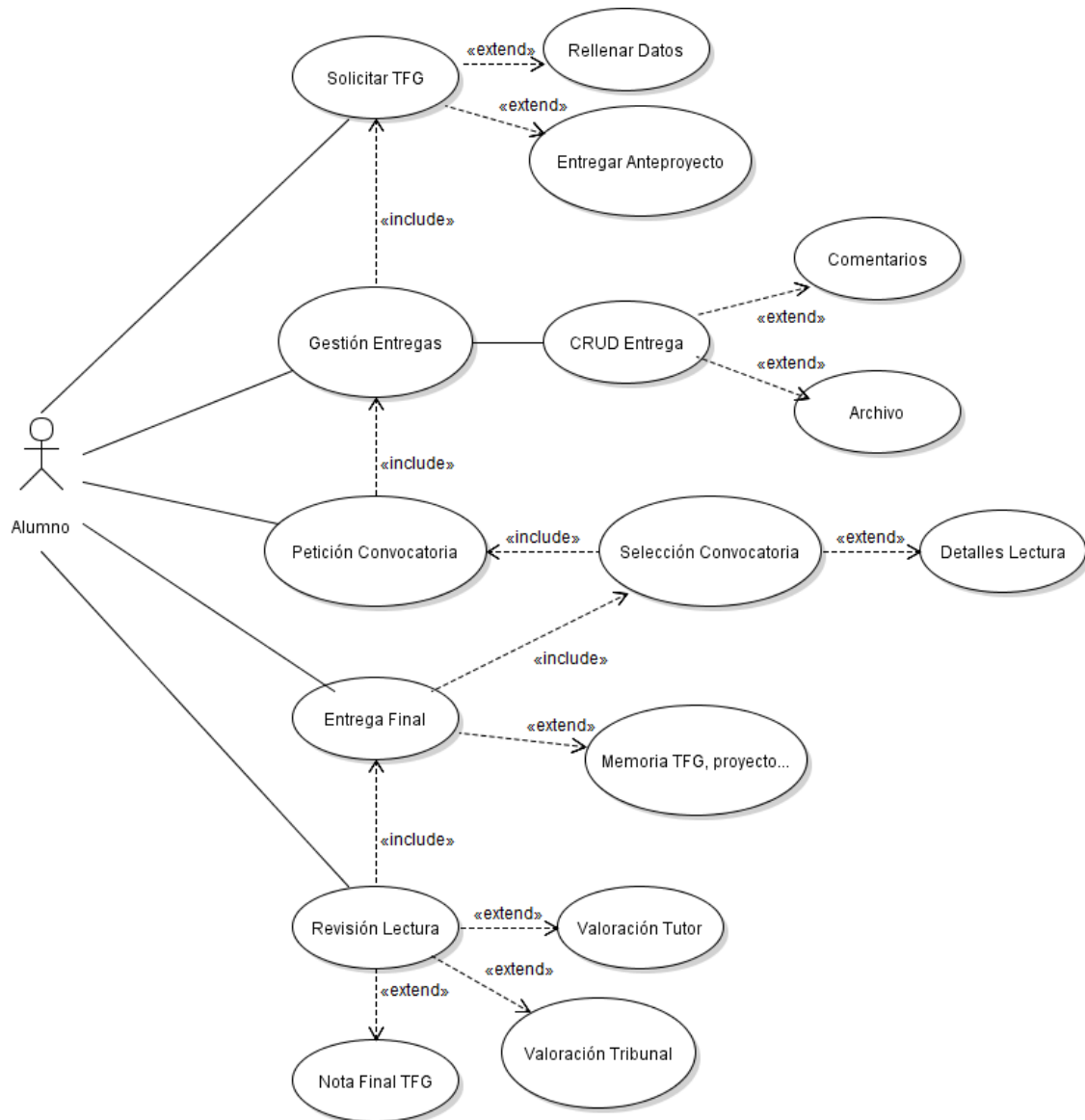


Figura 3. Diagrama de Casos de Uso – Rol Alumno.

Usuario Profesor

En el siguiente diagrama se muestran las funcionalidades que poseen todos los usuarios con rol profesor.

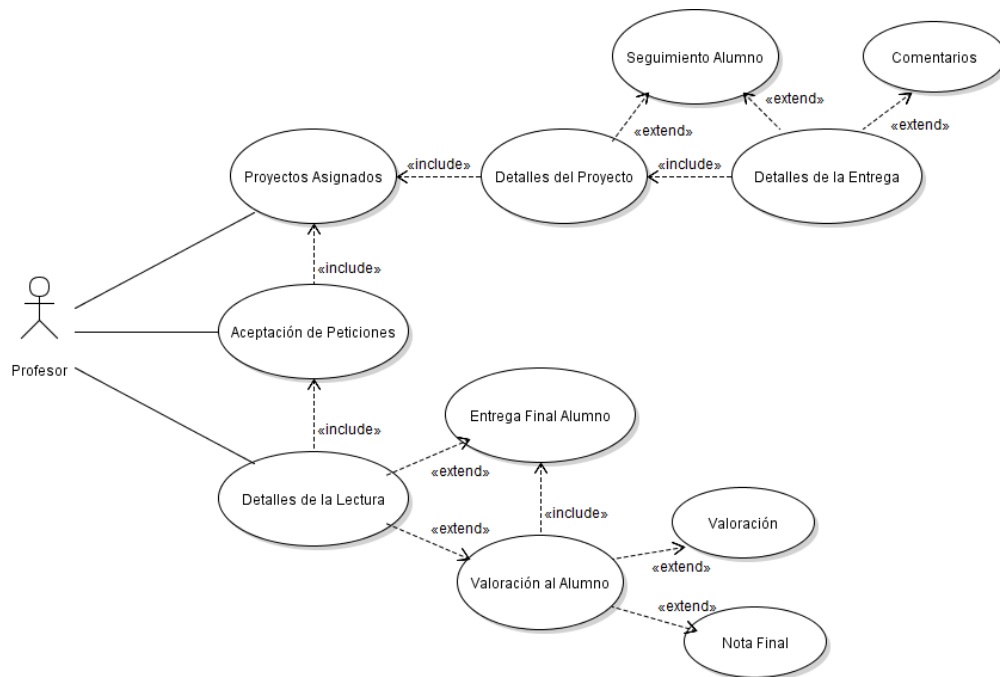


Figura 4. Diagrama de Casos de Uso – Rol Profesor.

Usuario Tribunal

En el siguiente diagrama se muestran las funcionalidades que poseen todos los usuarios con rol profesor, ya que un profesor puede ser un tribunal de otro proyecto.

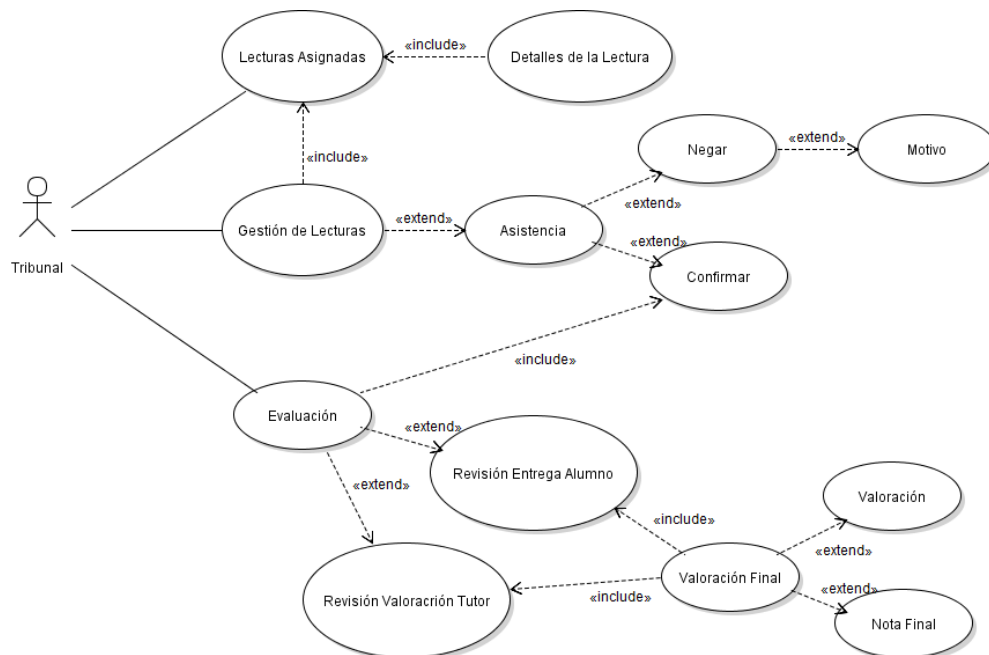


Figura 5. Diagrama de Casos de Uso – Rol Tribunal.

Usuario Administrador

En el siguiente diagrama se muestran las funcionalidades que poseen todos los usuarios con rol administrador.

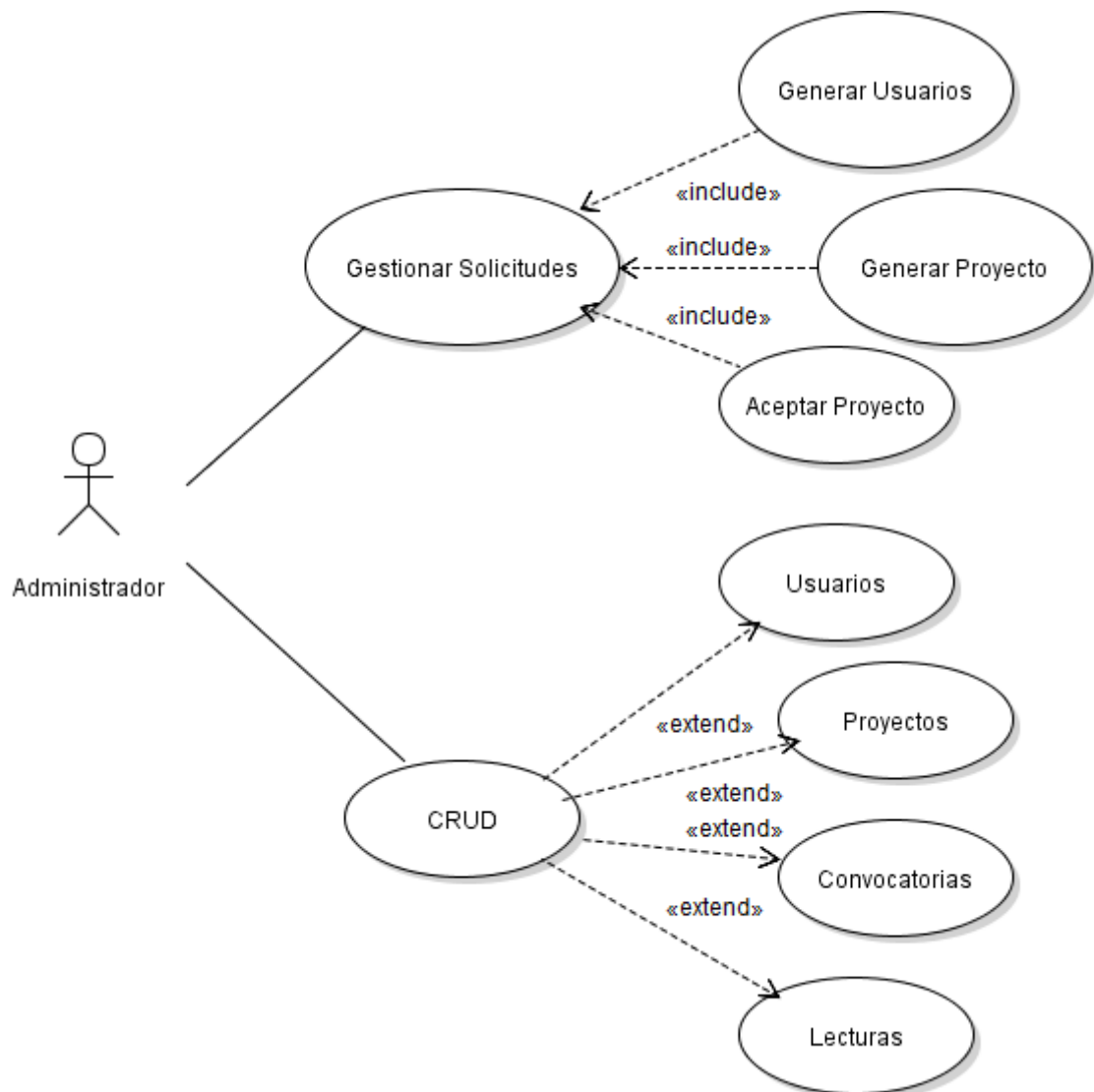


Figura 6. Diagrama de Casos de Uso – Rol Administrador.

Descripción de los actores

Usuario. Representa a todos los usuarios del sistema, independientemente de su rol. Todos los demás usuarios heredan las funcionalidades de éste.

Alumno. Representa a todos los alumnos de la ETSII que quieran realizar su TFG.

Profesor. Representa a todos los profesores de la ETSII, los cuales además de ser tutores de los TFG que tengan abiertos, podrán ser tribunal en las lecturas de otros TFG.

Tribunal. Representa a los profesores encargados de evaluar las presentaciones de los TFG.

Administrador. Representa a todo el personal de secretaría encargado de la gestión para la realización de un TFG.

Descripción de los Casos de Uso

Título	RF01. Identificación de usuario
Descripción	El usuario podrá acceder a la plataforma mediante una contraseña y su DNI
Pre-condición	Estar registrado en el Sistema
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario accede a la plataforma web, introduce su DNI y su contraseña. 2. El sistema valida al usuario y lo redirige a su página principal dependiendo del rol. 	
Escenario alternativo	
<ol style="list-style-type: none"> 2. El sistema no puede validar el inicio de sesión. 3. El sistema redirige al usuario a la página principal de la plataforma. 	
Clases de análisis	
A. Clases de entidad	Usuario
B. Clases de control	ctrlAcceso
C. Clases de interfaz	Index

Título	RF02. Notificaciones
Descripción	Cada usuario poseerá un panel de notificaciones donde se le avisará de cualquier cambio en el que esté involucrado.
Pre-condición	Estar registrado en el Sistema
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario inicia sesión en la plataforma web 2. El sistema carga las notificaciones del usuario y las muestra en el panel de notificaciones. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Notificacion
B. Clases de control	ctrlNotificaciones, main (FrontCtrl)
C. Clases de interfaz	main(Alumno,Profesor,Administrador), rellenarNotificaciones

Título	RF03. Detalles de un proyecto
Descripción	Cada proyecto tendrá una vista pública con sus datos.
Pre-condición	Estar registrado en el Sistema
Post-condición	
Prioridad	Media
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario inicia sesión en la plataforma web y accede al detalle de un proyecto. 2. El sistema recupera los datos del proyecto y los muestra en la vista de detalles. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Proyecto
B. Clases de control	detallarProyecto (FrontCtrl)
C. Clases de interfaz	detallarProyecto

Título	RF04. Perfil de usuario
Descripción	Cada usuario tendrá un panel donde ver y modificar sus datos personales. A excepción del DNI. Además, cada usuario tendrá un perfil público donde se muestren sus datos de contacto.
Pre-condición	Estar registrado en el Sistema
Post-condición	
Prioridad	Media
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario inicia sesión en la plataforma web y accede a su perfil. 2. El sistema carga los datos del usuario y los muestra. 	
Escenario alternativo	
<ol style="list-style-type: none"> 3. El usuario cambia sus datos (nombre, apellidos, email, foto, contraseña) y confirma los cambios. 4. El sistema guarda los datos en la base de datos. 	
Clases de análisis	
A. Clases de entidad	Usuario
B. Clases de control	ctrlPerfil
C. Clases de interfaz	adminPerfil, perfil (vista pública)

Título	RF05. Solicitar TFG
Descripción	Un alumno, una vez haya hablado con un tutor y haya decidido su TFG, rellenará una solicitud indicando los datos personales, del TFG y entregará en Anteproyecto.
Pre-condición	Haber concretado con el tutor el TFG y haber pedido en Secretaría la contraseña para generar la petición.
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El alumno accede a la plataforma, hace clic en solicitar TFG, rellena los datos necesarios y envía la solicitud. 2. El sistema procesa los datos, guarda la solicitud y redirige al usuario a una web donde se le notifica sobre la petición. 3. El alumno hace clic en inicio para volver al principio. 	
Escenario alternativo	
<ol style="list-style-type: none"> 2. El sistema notifica de algún fallo y redirige al alumno a su solicitud. 	
Clases de análisis	
A. Clases de entidad	Solicitud, Notificacion
B. Clases de control	ctrlSolicitud
C. Clases de interfaz	solicitar

Título	RF06. Gestión de Entregas
Descripción	El alumno podrá realizar entregas que forman parte de su proyecto, para que el tutor pueda tener un seguimiento del trabajo y corregir fallos. Podrá revisarlas y eliminarlas.
Pre-condición	Que su proyecto haya sido generado por la administración.
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El alumno accede a la plataforma y hace clic en realizar entrega. Rellena los datos necesarios sobre la entrega y hace clic en subir entrega. 2. El sistema procesa la entrega y la guarda en el Sistema. 3. El tutor es notificado. 	
Escenario alternativo	
<ol style="list-style-type: none"> 5. El alumno hace clic en una entrega, y hace clic en eliminar la entrega. 6. El sistema elimina la entrega. 5. El alumno hace clic en una entrega y revisa los comentarios del tutor. El alumno responde al comentario. 6. El sistema guarda el comentario y notifica al tutor. 	
Clases de análisis	
A. Clases de entidad	Proyecto, Entrega, Archivo, Notificacion, Comentario
B. Clases de control	ctrlEntregas, detallarEntrega(FrontCtrl), mainAlumno(FrontCtrl)
C. Clases de interfaz	mainAlumno, realizarEntrega, detallarEntrega
Diagramas de secuencia	
Diagrama de Secuencia – RF06. En el siguiente diagrama se muestra como un alumno realiza una entrega, accede a ella, realiza un comentario y la elimina (CRUD de entregas).	

Diagrama de Secuencia: RF06

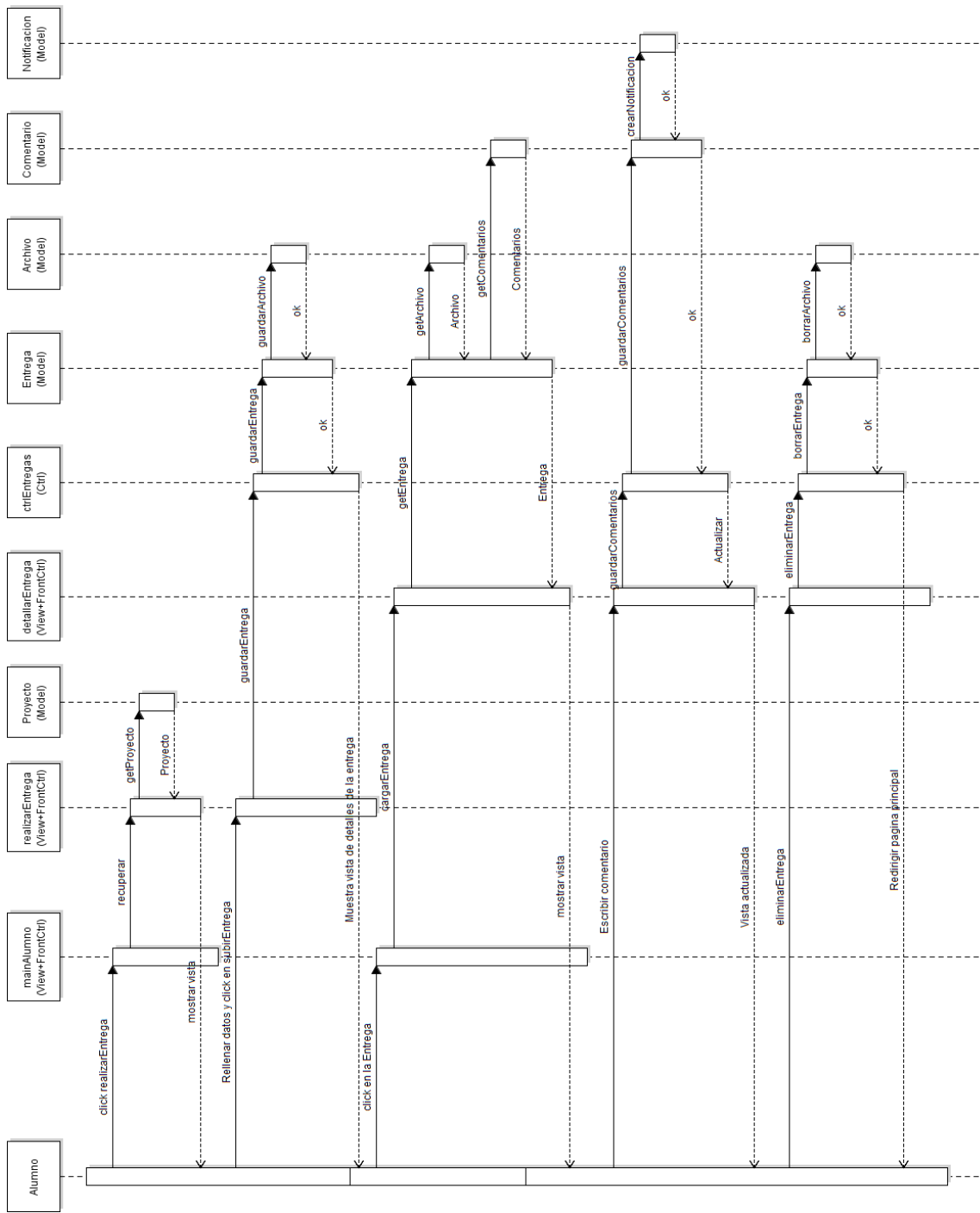


Figura 7. Diagrama de Secuencia – RF06.

Título	RF07. Petición de convocatoria
Descripción	El alumno tendrá una sección donde podrá pedir autorización al tutor para presentarse a una convocatoria. Una vez autorizado podrá presentarse a una de las convocatorias abiertas.
Pre-condición	Tener más de una entrega en el proyecto. Que el tutor haya autorizado el proyecto para presentar convocatoria.
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El alumno accede a la plataforma y hace clic en pedir autorización. 2. El sistema envía la petición al tutor. 3. El tutor acepta la petición. 4. El sistema notifica al alumno, y le da la posibilidad de seleccionar la convocatoria deseada 5. El alumno selecciona la convocatoria que desea. <p>Nota: Todos los usuarios implicados son debidamente notificados de cada paso.</p>	
Escenario alternativo	
<ol style="list-style-type: none"> 4. El tutor no acepta la autorización y se comunica con el alumno a través de su última entrega comunicándole los fallos. 5. El sistema guarda los comentarios y notifica al alumno. 	
Clases de análisis	
A. Clases de entidad	Proyecto, Notificacion, Convocatoria
B. Clases de control	ctrlPeticon, mainAlumno(FrontCtrl), solicitarConvocatoria(FrontCtrl)
C. Clases de interfaz	solicitarConvocatoria, mainAlumno
Diagramas de secuencia	
Diagrama de Secuencia – RF07. En el siguiente diagrama se muestra como un alumno realiza una petición de autorización a su tutor, y tras su autorización, realiza una selección de convocatoria.	

```

sequenceDiagram
    participant Alumno
    participant mainAlumno as mainAlumno (View-FrontCtrl)
    participant solicitarConvocatoria as solicitarConvocatoria (View-FrontCtrl)
    participant ctrlPetition as ctrlPetition (Ctrl)
    participant Proyecto as Proyecto (Model)
    participant Notificacion as Notificacion (Model)
    participant Convocatoria as Convocatoria (Model)
    participant Tutor

    Alumno->>mainAlumno: click pedirAutorizacion
    mainAlumno->>solicitarConvocatoria: pedirAutorizacion
    solicitarConvocatoria->>ctrlPetition: mostrar estado del proyecto
    ctrlPetition->>Proyecto: cambiarEstado
    Proyecto-->>ctrlPetition: Notificar tutor
    ctrlPetition->>Notificacion: ok
    ctrlPetition->>ctrlPetition: mostrar aviso y habilitar seleccionarConvocatoria
    ctrlPetition->>mainAlumno: seleccionarConvocatoria
    mainAlumno->>solicitarConvocatoria: cargarConvocatorias
    solicitarConvocatoria->>ctrlPetition: mostrar vista
    ctrlPetition->>mainAlumno: click en la convocatoria
    mainAlumno->>solicitarConvocatoria: guardarSelecion
    solicitarConvocatoria->>Convocatoria: getConvocatorias
    Convocatoria-->>ctrlPetition: Convocatorias
    ctrlPetition->>Proyecto: cambiarEstado
    Proyecto-->>ctrlPetition: ok
    ctrlPetition->>Notificacion: Notificar administracion
    Notificacion-->>ctrlPetition: ok
    solicitarConvocatoria->>mainAlumno: redirigir pagina principal
  
```

El alumno es notificado cuando el administrador ha creado su lectura

37

Título	RF08. Entrega final
Descripción	El alumno realizará una entrega final con la memoria y el proyecto de su TFG para su evaluación.
Pre-condición	Que se haya asignado su lectura.
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El alumno accede al detalle de la lectura, después hace clic en el detalle de la presentación y rellena los datos para la entrega final. Hace clic en guardar entrega. 2. El sistema guarda la entrega final. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Lectura, Proyecto, Presentacion
B. Clases de control	ctrlPresentacion, detallarLectura (FrontCtrl), detallarPresentacion (FrontCtrl)
C. Clases de interfaz	detallarLectura, detallarPresentacion

Título	RF09. Revisión de la presentación
Descripción	El alumno podrá acceder después de su lectura al detalle de la presentación y ver la calificación y la valoración del tutor y del tribunal.
Pre-condición	Haber realizado la entrega final y la lectura.
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El alumno realiza su lectura. 2. El tribunal califica al alumno y lo notifica. 3. El alumno accede al detalle de la presentación y observa la valoración y la calificación de su TFG. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Proyecto, Presentacion, Valoracion
B. Clases de control	ctrlPresentacion, detallarPresentacion (FrontCtrl)
C. Clases de interfaz	detallarPresentacion

Título	RF10. Proyectos asignados
Descripción	Panel donde el profesor tendrá acceso a los proyectos de los que es tutor y del estado en que se encuentra cada proyecto (pendiente de lectura, pendiente de autorización, ...)
Pre-condición	Que se le asigne el proyecto mediante la administración.
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El tutor accede en la plataforma y hace clic en uno de los proyectos de los que es tutor. 2. El Sistema redirige al tutor al panel de detalle del proyecto. 	
Escenario alternativo	
<ol style="list-style-type: none"> 2. El tutor hace clic en una de los avisos de un proyecto. 3. El Sistema redirige al tutor a la sección pertinente. 	
Clases de análisis	
A. Clases de entidad	Proyecto
B. Clases de control	mainProfesor (FrontCtrl)
C. Clases de interfaz	mainProfesor

Título	RF11. Detalles del proyecto
Descripción	Vista de los detalles de cada proyecto asignados al profesor. En esta vista podrá tener información acerca del proyecto y de las entregas que se realizan.
Pre-condición	Que se le asigne el proyecto mediante la administración.
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El tutor hace clic en un proyecto. 2. El Sistema carga los datos del proyecto desde la base de datos y los muestra de forma detallada junto con las entregas realizadas al tutor. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Proyecto
B. Clases de control	mainProfesor (FrontCtrl), detallarProyecto (FrontCtrl)
C. Clases de interfaz	detallarProyecto
Diagramas de secuencia	
Diagrama de Secuencia – RF11/RF12. En el siguiente diagrama de secuencia se muestra como un profesor accede a uno de los proyectos de los que es tutor, accede a una entrega del mismo, y realiza un comentario para el alumno.	

Título	RF12. Detalles de la entrega
Descripción	Panel donde se muestran los detalles de la entrega realizada por el alumno. En él, el profesor podrá revisar tanto los archivos subidos por el alumno, y poner comentarios a dicha entrega.
Pre-condición	El alumno realiza una entrega.
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El tutor accede a la vista de detalles de un proyecto y hace clic en una de las entregas. 2. El sistema carga la entrega, los comentarios, y el archivo desde la base de datos y muestra la vista de detalles de la entrega. 3. El profesor se descarga el archivo, lo valora y escribe un comentario al alumno. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Entrega, Archivo, Comentario, Notificacion
B. Clases de control	detallarEntrega (FrontCtrl)
C. Clases de interfaz	detallarEntrega
Diagramas de secuencia	
Diagrama de Secuencia – RF11/RF12. En el siguiente diagrama de secuencia se muestra como un profesor accede a uno de los proyectos de los que es tutor, accede a una entrega del mismo, y realiza un comentario para el alumno.	

Diagrama de Secuencia: RF11/12

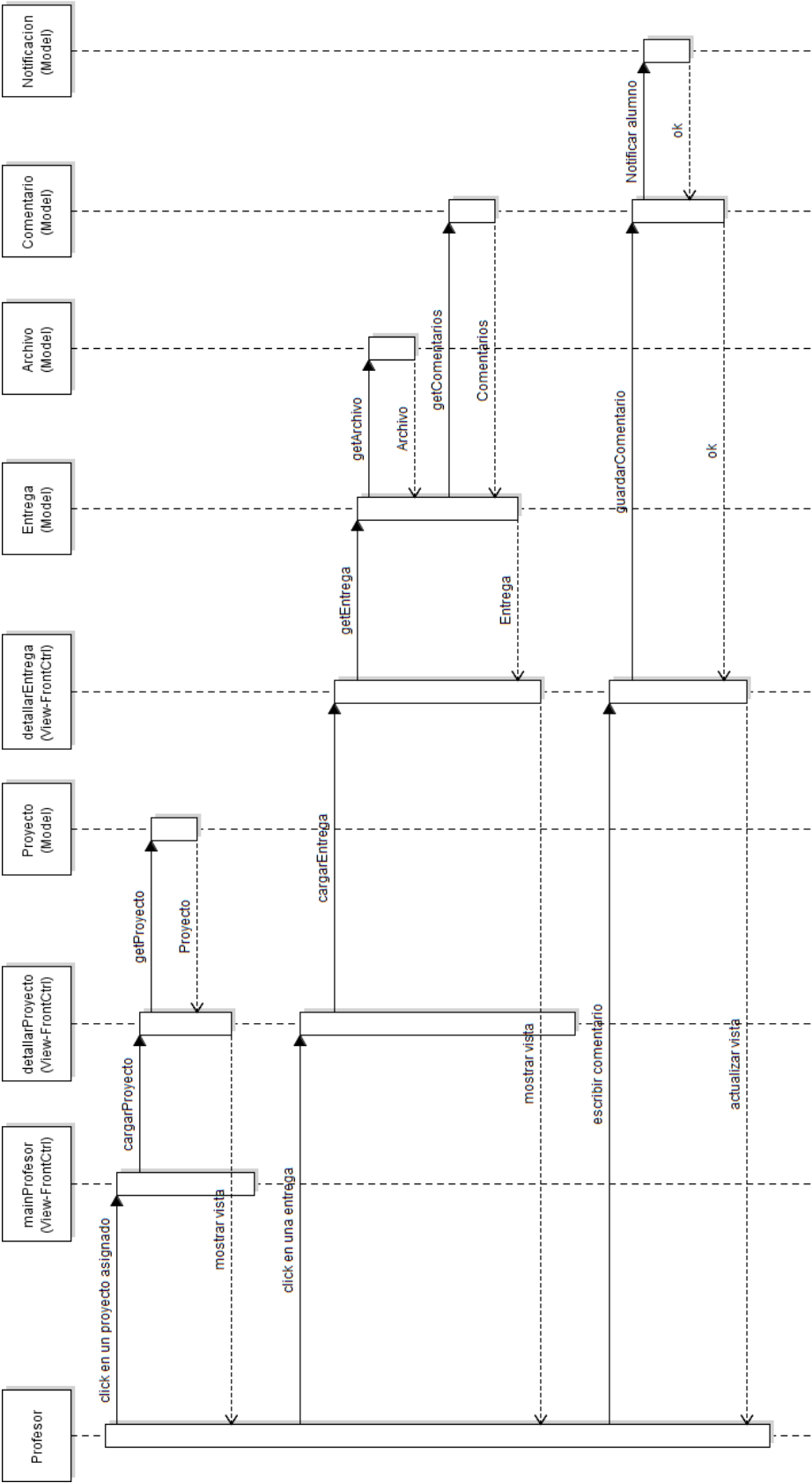


Figura 9. Diagrama de Secuencia – RF11/RF12.

Título	RF13. Autorización de proyectos
Descripción	El tutor de un proyecto será notificado de que el alumno de un proyecto quiere presentarse a convocatoria, éste deberá aceptar esta petición para seguir el proceso del alumno.
Pre-condición	El alumno pide la autorización para la convocatoria.
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El tutor accede a la notificación o al proyecto pendiente de la autorización. 2. El sistema carga la vista de detalles del proyecto y habilita un botón de autorización que el tutor puede hacer clic cuando crea que el proyecto está listo. 3. El tutor hace clic en aceptar solicitud. 4. El sistema guarda la autorización y notifica al alumno. 	
Escenario alternativo	
<ol style="list-style-type: none"> 4. El tutor no está de acuerdo con presentar convocatoria y comunica al alumno a través de los comentarios de la última entrega. 5. El sistema guarda el comentario. 	
Clases de análisis	
A. Clases de entidad	Proyecto, Notificacion
B. Clases de control	ctrlPeticon, detallarProyecto (FrontCtrl), mainProfesor(FrontCtrl)
C. Clases de interfaz	detallarProyecto, mainProfesor
Diagramas de secuencia	
Diagrama de Secuencia – RF13. En el siguiente diagrama se muestra como un tutor autoriza la petición de un alumno para la selección de convocatoria del proyecto.	

Diagrama de Secuencia: RF13

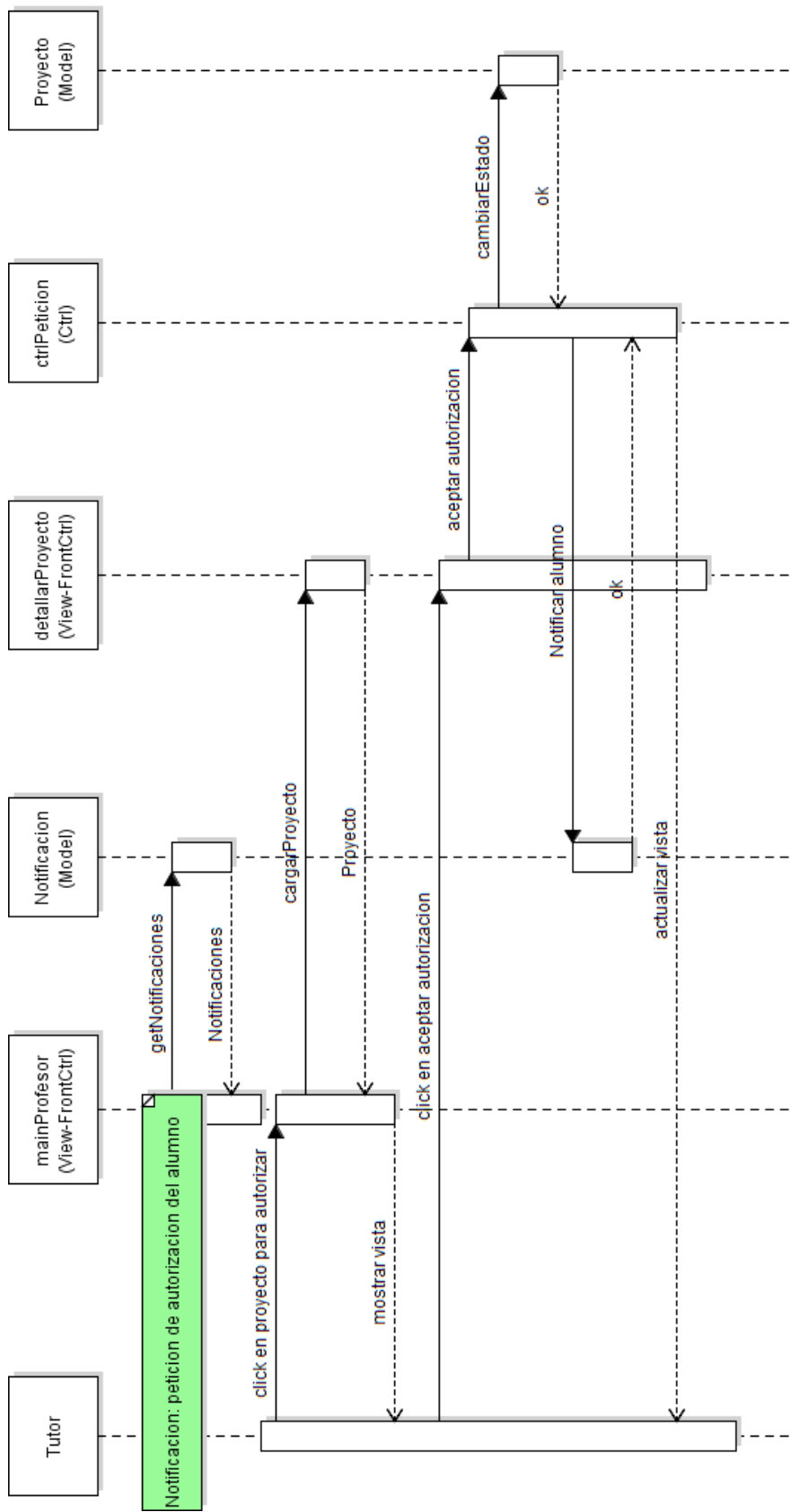


Figura 10. Diagrama de Secuencia – RF13.

Título	RF14. Valoración del alumnado
Descripción	El tutor de un proyecto poseerá un panel donde podrá valorar y calificar a sus alumnos.
Pre-condición	Se le deberá de haber asignado una lectura al TFG del alumno.
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El tutor accede en la plataforma y hace clic en la notificación de lectura asignada al proyecto. 2. El sistema carga la vista de detalle de la lectura. 3. El tutor hace clic en el botón de detalles de la presentación. 4. El sistema carga la presentación junto con los archivos que el alumno entregó y los muestra al tutor. 5. El tutor descarga los archivos, los valora y asigna una nota final. 6. El sistema guarda la valoración y la nota. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Presentacion, Valoracion, Notificacion, Archivo
B. Clases de control	ctrlPresentacion, detallarPresentacion(FrontCtrl), detallarLectura(FrontCtrl)
C. Clases de interfaz	detallarPresentacion, detallarLectura
Diagramas de secuencia	
Diagrama de Secuencia – RF14. En el siguiente diagrama se muestra como un tutor accede a la presentación final del alumno para realizar su valoración.	

Diagrama de Secuencia: RF14

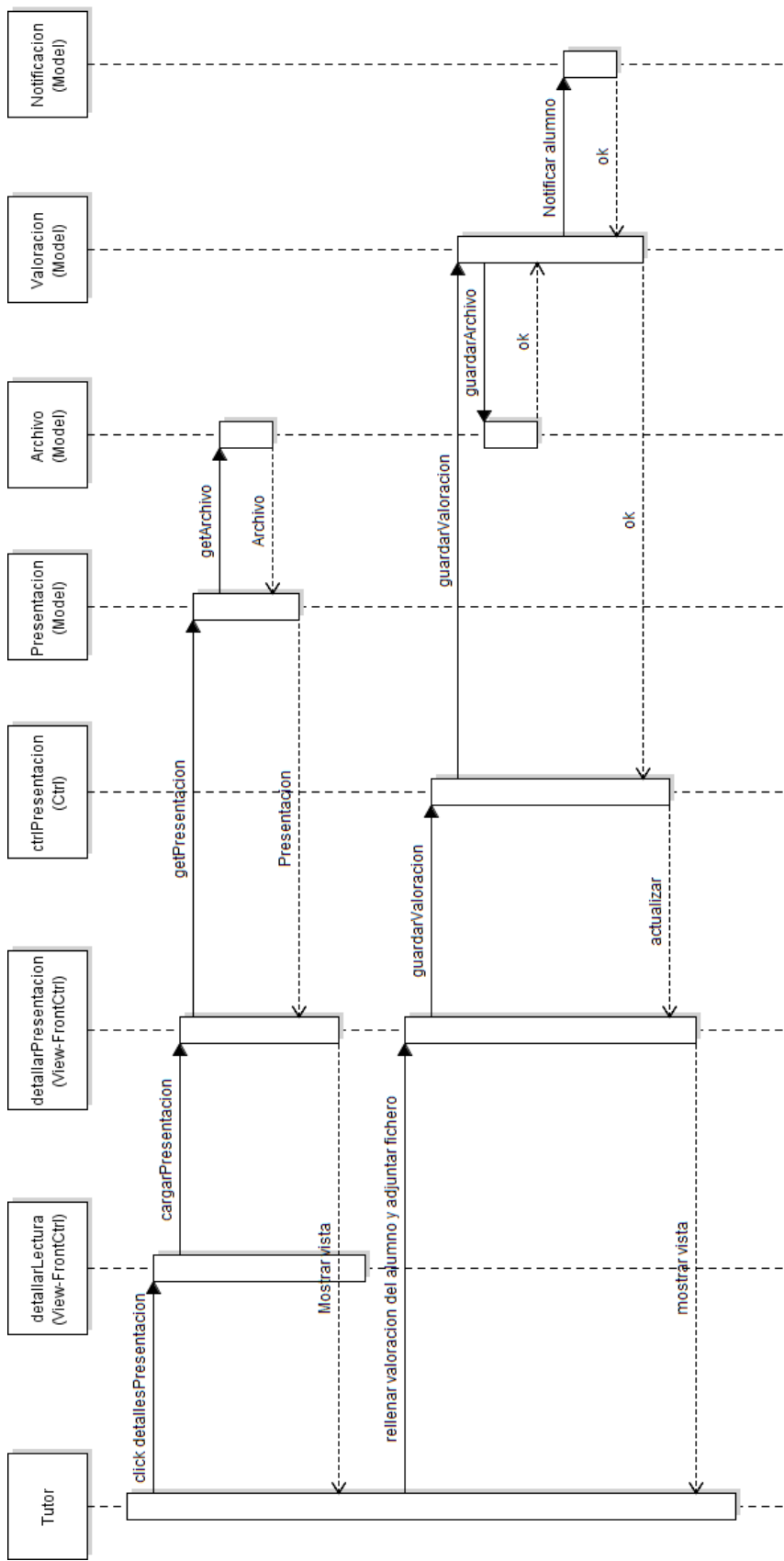


Figura 11. Diagrama de Secuencia – RF14.

Título	RF15. Lecturas asignadas
Descripción	El profesor tendrá un panel en el que ver las lecturas de los proyectos a los que asistirá como tribunal.
Pre-condición	Deberá haber sido asignado como tribunal de la lectura de un TFG.
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El profesor accede a la plataforma. 2. El sistema carga los datos de las lecturas que le fueron asignadas y que ha confirmado y las muestra en el panel principal junto con la fecha. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Lectura
B. Clases de control	mainProfesor (FrontCtrl)
C. Clases de interfaz	mainProfesor

Título	RF16. Gestión de lecturas
Descripción	El profesor tendrá un panel donde podrá pujar por las lecturas que se encuentren abiertas. El administrador se hará cargo de seleccionar a los miembros del profesorado que hayan pujado “Sí” preferentemente o “No” en caso contrario como miembros del tribunal.
Pre-condición	
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El profesor accede al panel de gestión de lecturas. 2. El sistema carga las lecturas que se encuentren abiertas. 3. El profesor hace clic en la lectura que aún no haya pujado. 4. El sistema muestra un modal para seleccionar la respuesta. 5. El profesor hace clic en el botón de “Sí” para confirmar su interés por la lectura y hace clic en guardar. 6. El sistema guarda la respuesta del profesor. <p>Nota: El profesor tendrá acceso a la vista de detalles del proyecto desde el panel de asistencia.</p>	
Escenario alternativo	
<ol style="list-style-type: none"> 6. El profesor hace clic en el botón de “No” para indicar su falta de idoneidad como tribunal. 7. El sistema le pide que indique el motivo. 8. El profesor indica el motivo. 9. El sistema guarda la respuesta del profesor y notifica a la administración. 	
Clases de análisis	
A. Clases de entidad	Lectura, Proyecto, Tribunal
B. Clases de control	ctrlTribunal, pujarLecturas(FrontCtrl), mainProfesor(FrontCtrl), detallarProyecto(FrontCtrl)
C. Clases de interfaz	pujarLecturas, mainProfesor, detallarProyecto
Diagramas de secuencia	
Diagrama de Secuencia – RF16. En el siguiente diagrama se muestra como un profesor puja en las lecturas que se encuentran abiertas.	

Diagrama de Secuencia: RF16

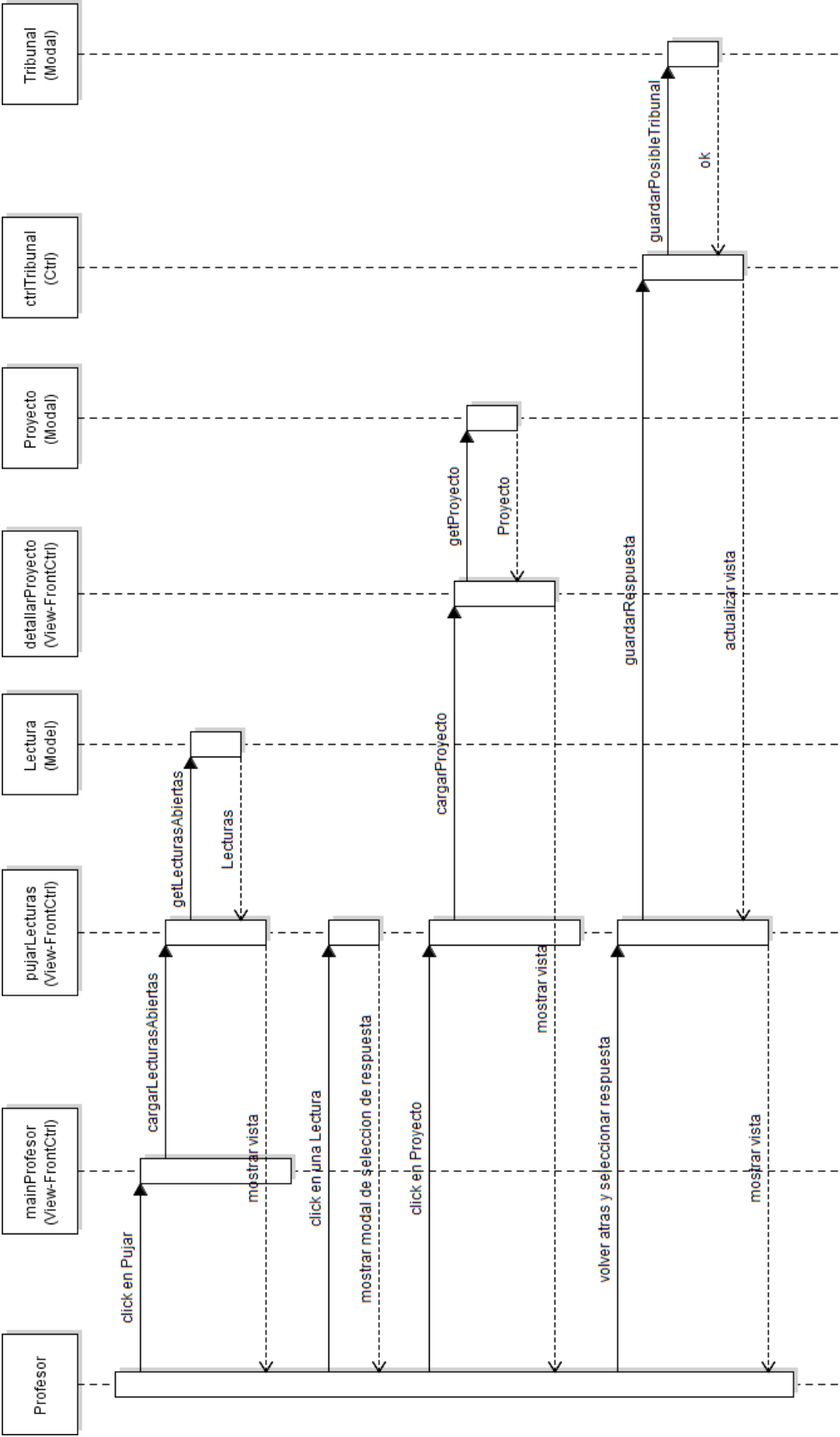


Figura 12. Diagrama de Secuencia – RF16.

Título	RF17. Detalles de la lectura
Descripción	El profesor que puge y sea asignado a una lectura tendrá acceso a ella, donde podrá ver los detalles, el tribunal asignado, día, hora, ...
Pre-condición	Pujar sobre una lectura y ser asignado a ella.
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El profesor accede en la plataforma y hace clic en una de las lecturas de su panel de lecturas. 2. El sistema carga los datos de los detalles de la lectura. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Lectura, Proyecto, Tribunal
B. Clases de control	detallarLectura (FrontCtrl)
C. Clases de interfaz	detallarLectura

Título	RF18. Evaluación
Descripción	El profesor que asiste como tribunal a una lectura tendrá acceso a un panel donde podrá revisar la entrega final del alumno, y la valoración que realizó el tutor. Si el profesor fue asignado como Presidente podrá en conjunto con el Secretario y el Vocal asignar una valoración y una calificación final al alumno.
Pre-condición	
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El profesor accede a los detalles de una lectura y hace clic en el botón de detalles de la presentación. 2. El sistema carga los datos de la presentación (Entrega del alumno y valoración del tutor). 3. El profesor (presidente) junto con los otros profesores miembros del tribunal rellenan un panel donde calificar al alumno. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Proyecto, Presentacion, Valoracion, Archivo, Notificacion
B. Clases de control	ctrlPresentacion, detallarPresentacion(FrontCtrl), detallarLectura(FrontCtrl)
C. Clases de interfaz	detallarPresentacion, detallarLectura
Diagramas de secuencia	
Diagrama de Secuencia – RF18. En el siguiente diagrama se muestra como un profesor, que asiste como tribunal a una lectura, accede a los detalles de la última entrega del alumno, revisa la entrega y la valoración del tutor, y realiza una valoración final acerca del TFG.	

Diagrama de Secuencia: RF18

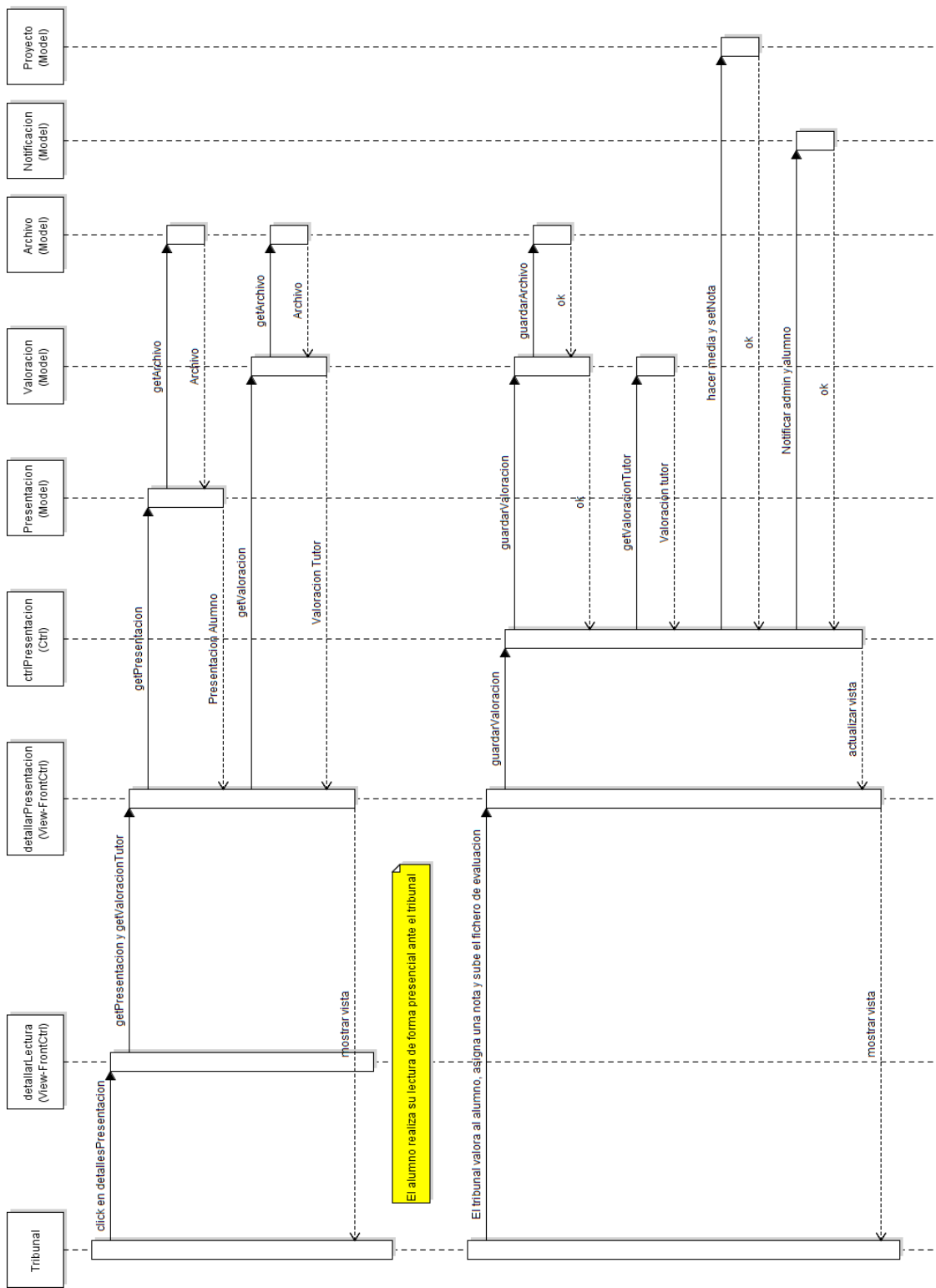


Figura 13. Diagrama de Secuencia – RF18.

Título	RF19. Gestión de solicitudes
Descripción	El administrador tendrá acceso a las solicitudes que realicen los alumnos. En esta sección podrá generar los usuarios para los alumnos y su proyecto.
Pre-condición	Petición realizada por el alumno
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador accede a la plataforma, y hace clic en el apartado de solicitudes. 2. El sistema recupera las solicitudes pendientes del sistema y las muestra en una lista al administrador. 3. El administrador hace clic en la solicitud que desee gestionar. 4. El sistema recupera la información acerca de esa solicitud y la muestra al administrador. 5. El administrador descarga los archivos de la solicitud (Anteproyecto) y hace clic en validar y generar proyecto. 6. El sistema genera los usuarios y los vincula al proyecto, luego cambia el estado de la solicitud a aceptada y redirige al administrador al panel de solicitudes pendientes. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Solicitud, Archivo, Usuario, Proyecto, Profesor, Alumno
B. Clases de control	ctrlSolicitudes, detallarSolicitud(FrontCtrl), mainAdmin(FrontCtrl),
C. Clases de interfaz	detallarSolicitud, mainAdmin
Diagramas de secuencia	
Diagrama de Secuencia – RF19. En el siguiente diagrama se muestra como un administrador accede a las solicitudes de TFG, genera el usuario para el alumno y genera el proyecto.	

Diagrama de Secuencia: RF19

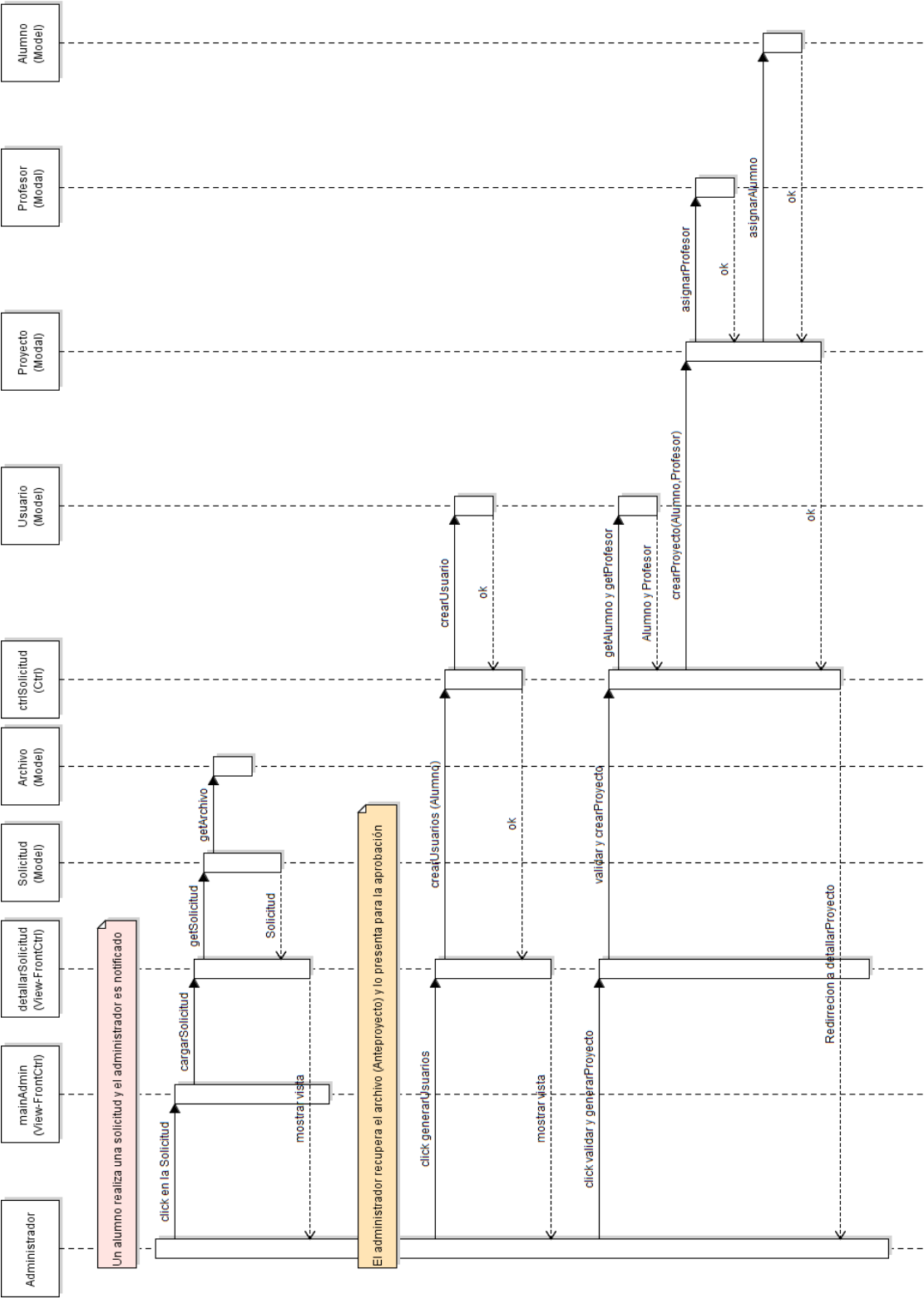


Figura 14. Diagrama de Secuencia – RF19.

Título	RF20. CRUD Usuarios
Descripción	El administrador tendrá un panel donde ver, crear, modificar y eliminar cualquier usuario del sistema.
Pre-condición	
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador accede a la plataforma, y hace clic en el apartado de usuarios. 2. El sistema carga los datos de los usuarios del sistema, los muestra y da al administrador la posibilidad de ver, crear, modificar o eliminar un usuario. 3. El administrador selecciona la opción que desea realizar. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Usuario
B. Clases de control	ctrlUsuarios, addUsuario(FrontCtrl), adminUsuarios (FrontCtrl), perfilUsuario(FrontCtrl)
C. Clases de interfaz	addUsuario, adminUsuarios, perfilUsuario

Título	RF21. CRUD Proyectos
Descripción	El administrador tendrá un panel donde ver, crear, modificar y eliminar cualquier proyecto del sistema.
Pre-condición	
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador accede a la plataforma, y hace clic en el apartado de proyectos. 2. El sistema carga los datos de los proyectos del sistema, los muestra y da al administrador la posibilidad de ver, crear, modificar o eliminar un proyecto. 3. El administrador selecciona la opción que desea realizar. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Proyecto, Usuario
B. Clases de control	ctrlProyectos, addProyecto(FrontCtrl), adminProyectos(FrontCtrl), detallarProyecto(FrontCtrl)
C. Clases de interfaz	addProyecto, adminProyectos, detallarProyecto

Título	RF22. CRUD Convocatorias
Descripción	El administrador tendrá un panel donde ver, crear, modificar y eliminar cualquier convocatoria del sistema.
Pre-condición	
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador accede a la plataforma, y hace clic en el apartado de convocatorias. 2. El sistema carga los datos de las convocatorias del sistema, los muestra y da al administrador la posibilidad de ver, crear, modificar o eliminar una convocatoria. 3. El administrador selecciona la opción que desea realizar. 	
Escenario alternativo	
Clases de análisis	
A. Clases de entidad	Convocatoria
B. Clases de control	ctrlConvocatorias, addConvocatoria(FrontCtrl), adminConvocatorias(FrontCtrl), detallarConvocatoria(FrontCtrl)
C. Clases de interfaz	addConvocatoria, adminConvocatoria, detallarConvocatoria

Título	RF23. CRUD Lecturas
Descripción	El administrador tendrá un panel donde ver, crear, modificar y eliminar cualquier lectura del sistema.
Pre-condición	
Post-condición	
Prioridad	Alta
Autor(es)	Juan José Ariza Fernández
Control de cambios	V 1.0
Escenario principal	
<ol style="list-style-type: none"> 1. El administrador accede a la plataforma, y hace clic en el apartado de convocatorias, y luego en cada convocatoria tiene acceso a las lecturas diferenciadas por días. 2. El sistema carga los datos de las convocatorias del sistema, los muestra y da al administrador la posibilidad de ver, crear, modificar o eliminar una lectura. 3. El administrador selecciona la opción que desea realizar. 	
Escenario alternativo	
<ol style="list-style-type: none"> 3. El administrador accede a una lectura para asignar el tribunal. 4. El sistema muestra los profesores que pujaron por la lectura. 5. El administrador hace clic en añadir en el profesor y selección el rol que desee. 6. El sistema guarda el tribunal y lo notifica. 	
Clases de análisis	
A. Clases de entidad	Lectura, Tribunal, Notificación, Proyecto
B. Clases de control	ctrlLecturas, addLectura(FrontCtrl), detallarLectura(FrontCtrl)
C. Clases de interfaz	addLectura, detallarLectura
Diagramas de secuencia	
Diagrama de Secuencia – RF23. En el siguiente diagrama se muestra como un administrador accede a una lectura y añade un tribunal.	

Diagrama de Secuencia: RF23

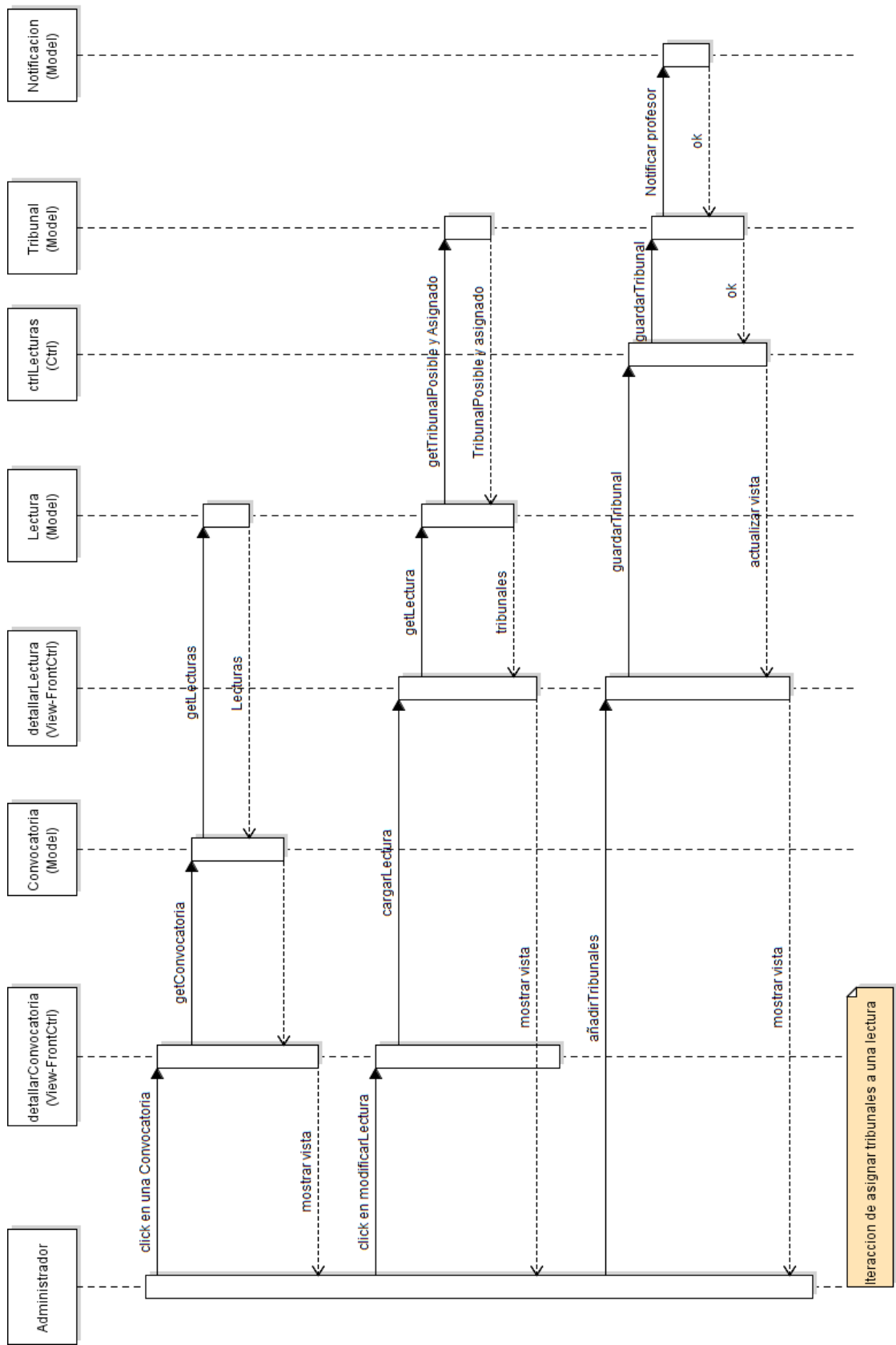


Figura 15. Diagrama de Secuencia – RF23.

5. Diseño del Sistema

Diseño Lógico de Datos

Un diagrama de clases de diseño recoge tanto los conceptos del dominio del sistema propios del diagrama de clases de análisis como aquellos conceptos que forman parte de la implementación de la propia aplicación del Sistema. Una clase del modelo de diseño, además de representar un concepto del mundo real en el que actúa el Sistema, también puede representar un concepto de implementación del sistema que se está modelando.

Dominio de Negocio (Modelo de Datos)

En el siguiente diagrama se muestra el diseño de las tablas de la base de datos, éstas se harán cargo de almacenar toda la información perteneciente a la plataforma web.

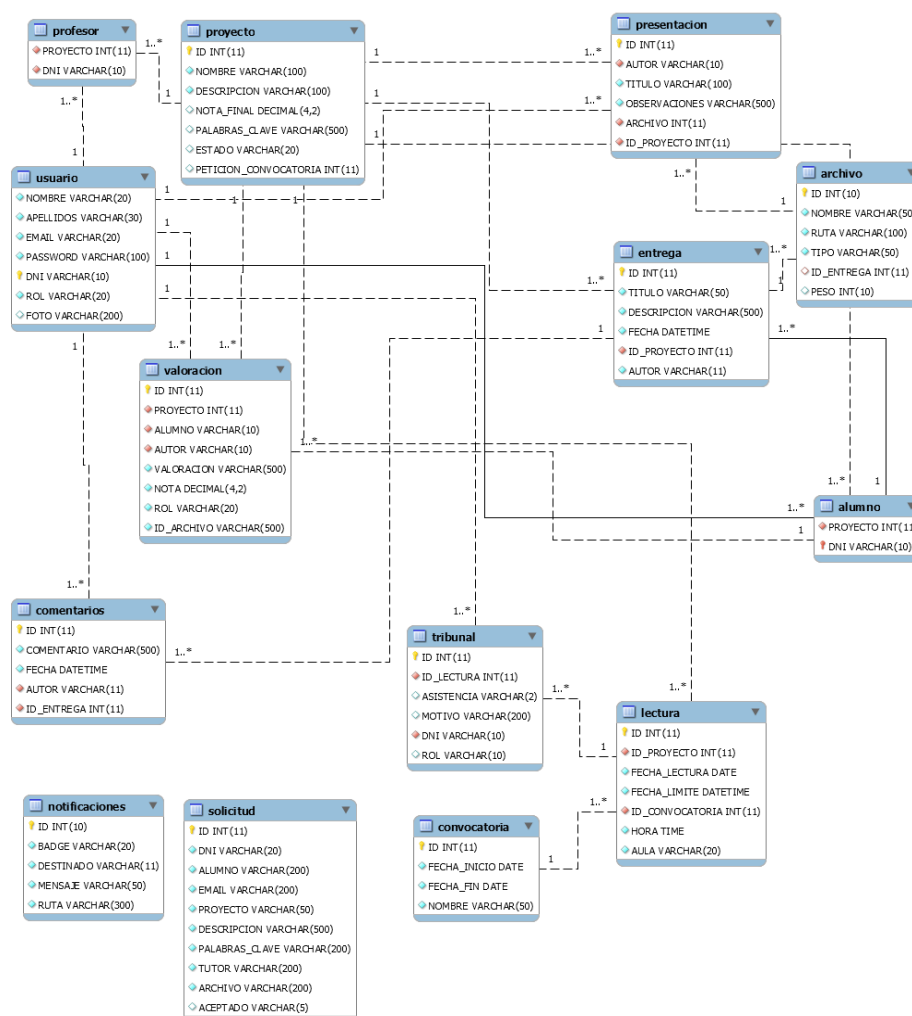


Figura 16. Diagrama de la Base de Datos.

Clases de Datos

En el siguiente diagrama se muestra el diseño de las clases codificadas en PHP. Estas harán de la capa de modelo, y se harán cargo de recuperar la información necesaria de la base de datos y construir el objeto correspondiente para ser usado por el controlador.

Cada una de estas clases estará compuesta por métodos “get” para cada uno de sus atributos, métodos para crear, ver, modificar o eliminar información perteneciente a la clase en la base de datos, y una serie de métodos propios de cada clase enfocados a diferentes procesos.

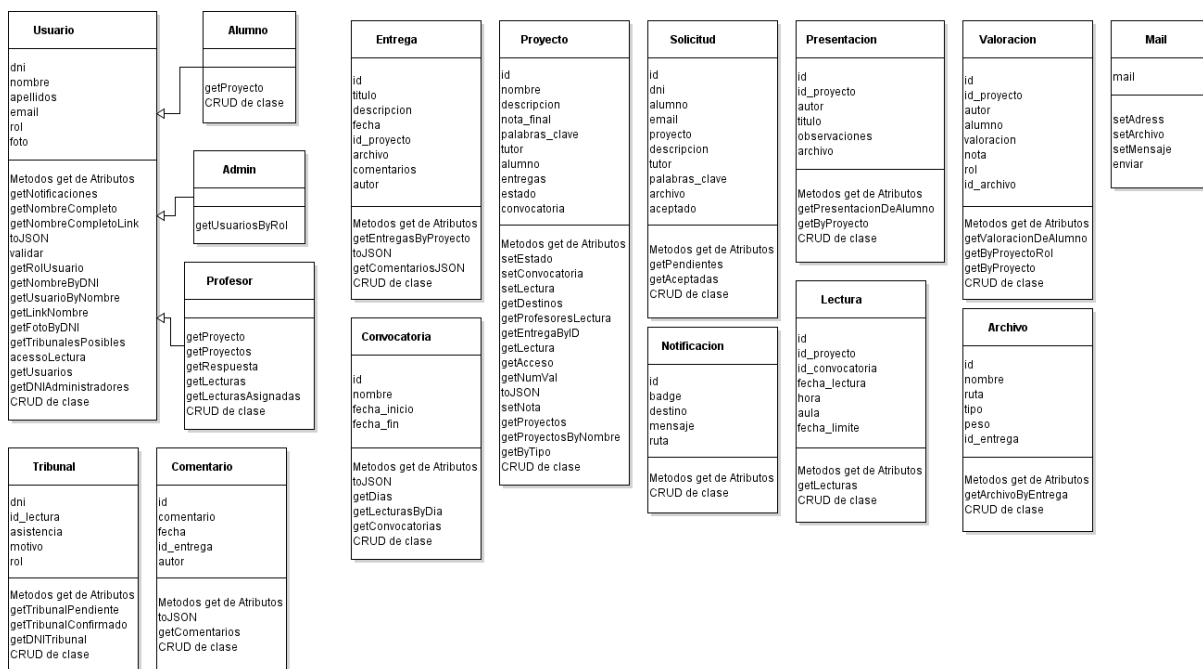


Figura 17. Diagrama de clases.

Clases del Controlador

En el siguiente diagrama se muestra el diseño de los controladores pertenecientes a cada sección de la plataforma, cada uno de ellos está compuesto por una serie de métodos para intercambiar información con la capa de modelo.

Todas las vistas están compuestas de un controlador propio(FrontCtrl), el cual se encarga de comprobar que el usuario haya iniciado sesión, que su rol sea el adecuado para ese recurso, que tenga acceso al recurso y de proporcionar la información necesaria para esa vista.

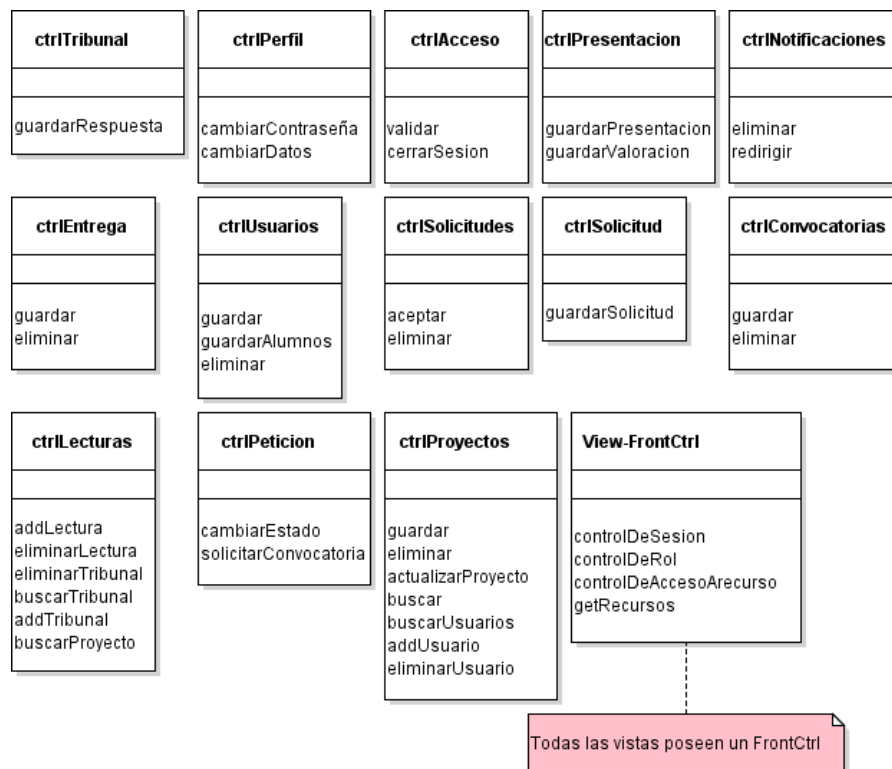


Figura 18. Diagrama de controladores.

Clases de Servicios

En el siguiente diagrama se muestra las vistas que se harán cargo de mostrar la información al usuario.

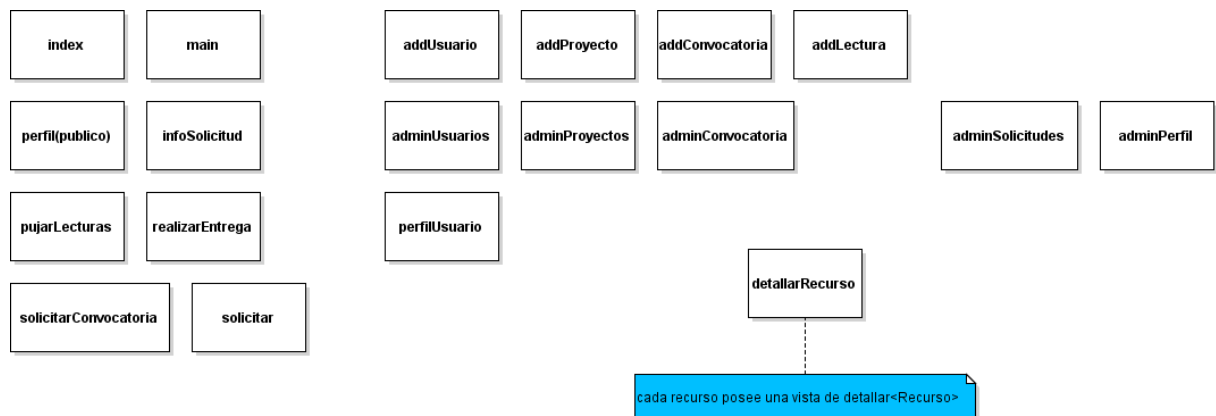


Figura 19. Diagrama de vistas.

Diseño Físico de Datos

A partir del diseño físico de datos se pretende representar y definir todos los datos que se introducen, almacenan, transforman y producen dentro del sistema de información. Dado que el modelo de datos es un medio para comunicar el significado de los datos y las relaciones entre ellos, ofrece una serie de ventajas que son las siguientes:

- Comprensión de los datos de una organización y de su funcionamiento.
- Obtención de estructuras de datos independientes del entorno físico.
- Control de los posibles errores desde el principio, o al menos, darse cuenta de las deficiencias lo antes posible.
- Mejora del mantenimiento del Sistema.

Aunque la estructura de datos puede ser cambiante y dinámica, normalmente es mucho más estable que la estructura de procesos. Como resultado, una estructura de datos estable e integrada proporciona datos consistentes que puedan ser fácilmente accesibles según las necesidades de los usuarios, de manera que, aunque se produzcan cambios organizativos, los datos permanecerán estables.

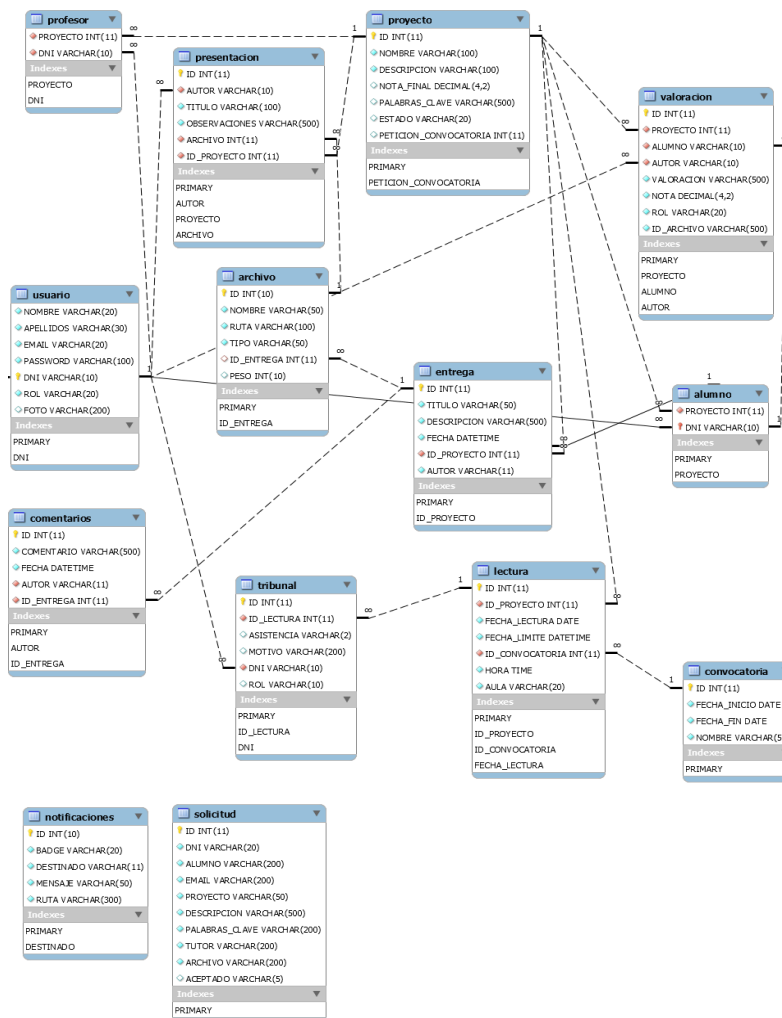


Figura 20. Diagrama físico de la base de datos.

6. Implementación

Como se ha explicado en el presente documento, el desarrollo del proyecto tiene como objetivo la creación de una plataforma web para la gestión de los TFG. A continuación, se explicará cómo se ha llevado a cabo y qué decisiones se han tomado. Además, se podrá observar de manera global como funciona, es decir, como interaccionan cada una de las capas.

Detalle de las tecnologías

En puntos anteriores se ha descrito el entorno tecnológico utilizado. En esta sección se detallará las versiones aplicadas de cada una de estas tecnologías.

Herramientas

- Sublime Text 3 como editor de textos
- XAMPP en su versión 5.6.20 (Control Panel 3.2.2)
- phpMyAdmin versión 4.5.1
- MySQL Workbench versión 6.3
- SourceTree versión 1.9.6.1 para el control de desarrollo
- Google Chrome e inspector para el desarrollo de las interfaces

PHP

- PHP versión 5.6.20
- Extensión mysqli
- PHPMailer versión 5.2.16

MySQL

- MySQL versión 5.6 distribución 10.1.13-MariaDB

JavaScripts

- Distribución Bootstrap 3.3.7 (opensource)
 - Bootstrap.min.js
- jquery-1.12.3.min.js (opensource)
- jquery-ui.min.js versión 1.12.0 (opensource)
- jquery.timepicker.min.js (opensource)

CSS

- Distribución Bootstrap 3.3.7
 - bootstrap.min.css
 - bootstrap-theme.min.css
- Distribución JQuery 1.12.3
 - jquery-ui.min.css
 - jquery-ui.theme.min.css

Preparación del proyecto

Para la creación de dicho proyecto, en primer lugar, descargaremos e instalaremos XAMPP.

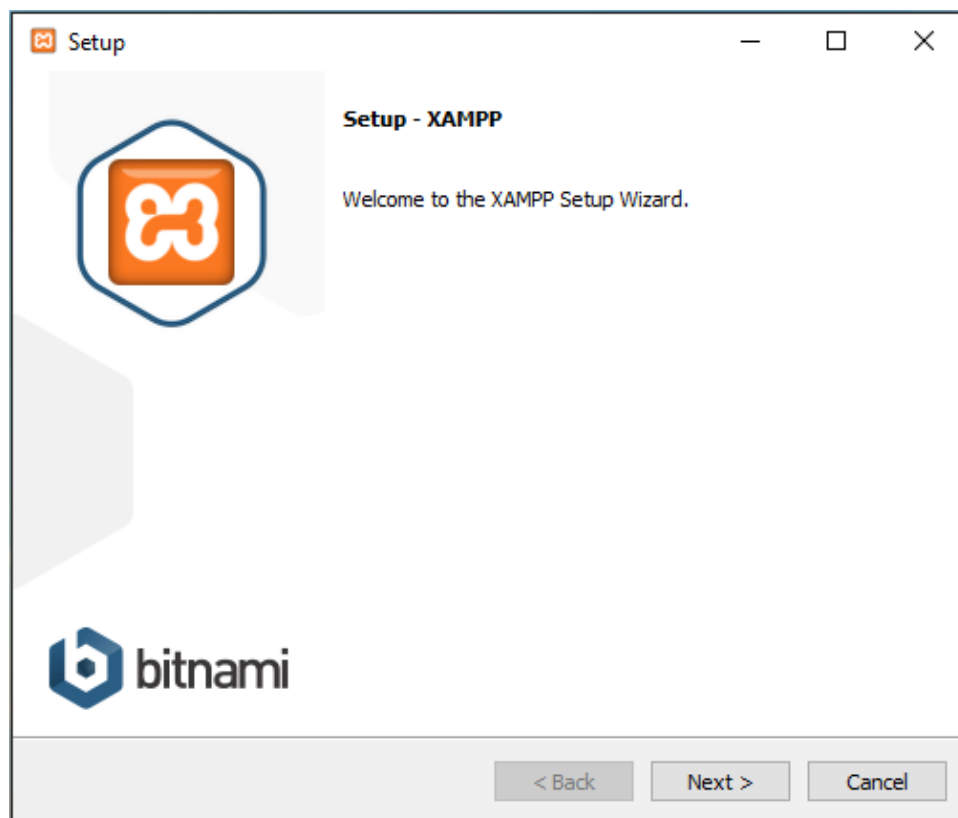


Figura 21. Instalación XAMPP.

Una vez que hemos acabado la instalación de XAMPP, abrimos el panel de control y arrancamos el servidor Apache y MySQL.

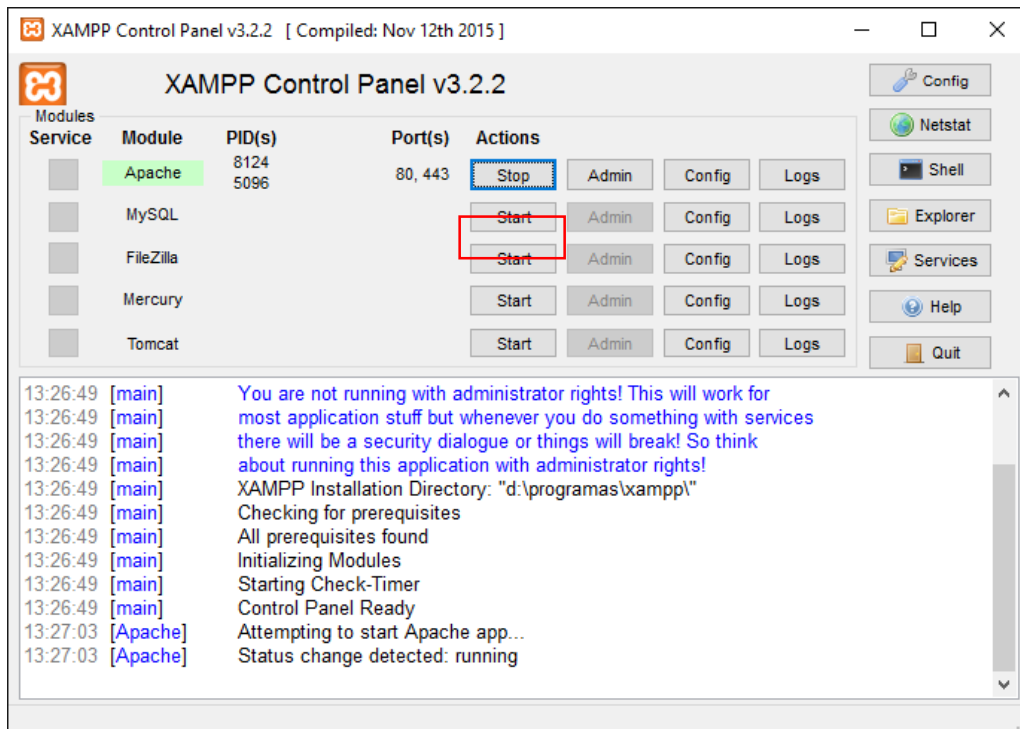


Figura 22. Panel XAMPP.

Una vez haya arrancado el servidor Apache y MySQL, ya tendremos acceso a él ingresando la url “http://localhost/” en el navegador.

El siguiente paso a seguir es crear la carpeta de nuestro proyecto en el directorio htdocs de XAMPP, para que éste pueda reconocerlo y darnos acceso a él. Para ello vamos a la carpeta de instalación de XAMPP y buscamos la carpeta htdocs.

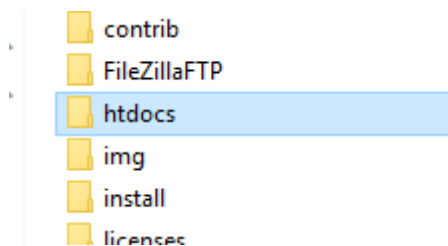


Figura 23. Carpeta de alojamiento de la web en XAMPP.

Una vez hemos ingresado, creamos una nueva carpeta, la cual será nuestro proyecto, y al que podremos acceder mediante la siguiente url “http://localhost/<nombre_nueva_carpeta>”, que en mi caso es “http://localhost/tfg”.

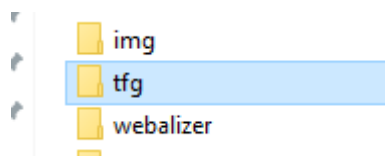


Figura 24. Carpeta de la web.

Una vez creada la carpeta de nuestro proyecto, ingresamos en ella, y será aquí donde iremos creando tanto los archivos como los ficheros con extensión php de nuestra web. Para una mayor facilidad a la hora de la programación arrastraremos la carpeta a Sublime Text, ya que este programa dispone de un gestor de archivos, y nos será más fácil poder gestionar la creación de la web.

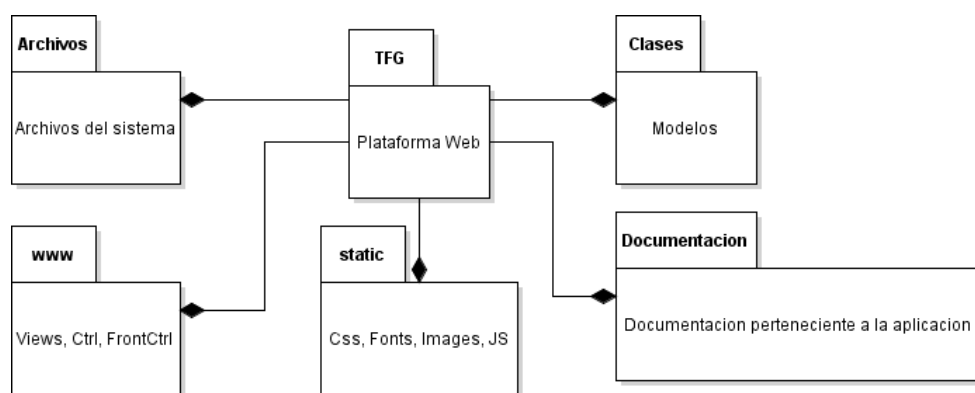


Figura 25. Estructura de paquetes.

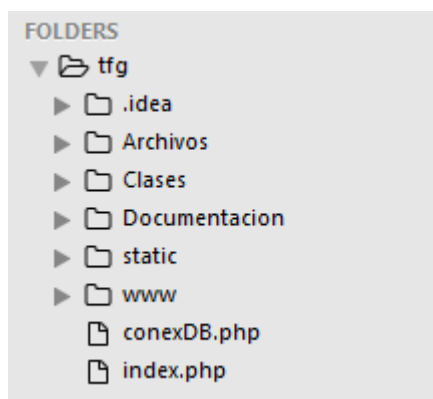


Figura 26. Estructura de paquetes en sublime text.

El primer paso para comenzar con la creación de la web será crear una página de inicio (index), la cual, será detectado por XAMPP y la mostrará como página principal al ingresar en la url de nuestro proyecto.

El siguiente paso será crear las diferentes carpetas para estructurar la aplicación, en nuestro caso lo haremos de la siguiente forma:

Archivos. Carpeta encargada de guardar los archivos subidos por los usuarios.

Clases. Carpeta encargada de guardar las clases de PHP y que se harán cargo de recuperar los datos almacenados en la Base de Datos.

Documentacion. Carpeta donde alojaremos archivos o documentos relacionados con la plataforma para una mejor comprensión de la misma.

Static. Carpeta encargada de los archivos estáticos de la web.

www. Carpeta encargada de guardar la aplicación, ya sean las vistas como los controladores.

conexBD. Archivo que se encarga de la conexión con la Base de Datos.

Index. Archivo donde se encuentra definido la página principal.

Una vez hemos instalado y estructurado la aplicación en la carpeta de XAMPP, procedemos a preparar la Base de Datos para poder hacer conexiones con ella.

En primer lugar, entramos en phpMyAdmin mediante el panel de control de XAMPP. Una vez dentro creamos una nueva Base de Datos para nuestro proyecto.

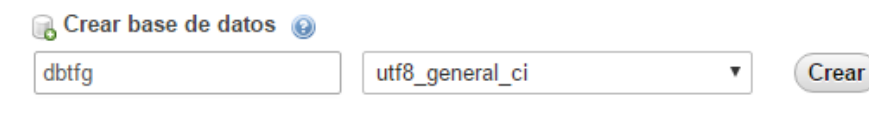


Figura 27. Creación de la base de datos.

Cuando ya hemos creado la Base de Datos, accedemos a ella, y creamos las tablas necesarias para guardar los datos de nuestra aplicación. En nuestro caso vamos a importar un archivo donde se encuentran todas las tablas preparadas.

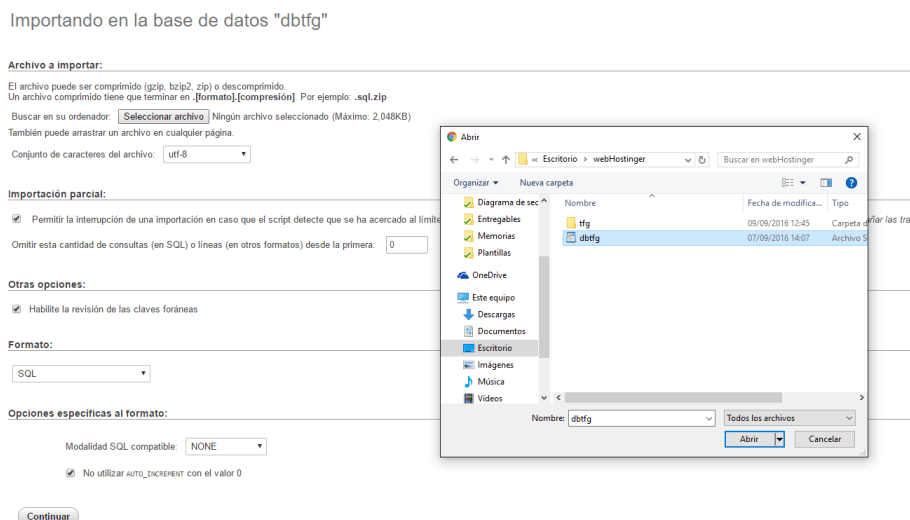


Figura 28. Importar base de datos.

Una vez importado el archivo, hacemos clic en continuar y se generarán nuestras tablas junto con sus relaciones. Además, se generará un usuario Administrador para poder gestionar el sistema. Por tanto, una vez importado el archivo veremos que nuestra Base de Datos queda así:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> alumno	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	32 KB	-
<input type="checkbox"/> archivo	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	32 KB	-
<input type="checkbox"/> comentarios	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	48 KB	-
<input type="checkbox"/> convocatoria	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	16 KB	-
<input type="checkbox"/> entrega	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	48 KB	-
<input type="checkbox"/> lectura	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	64 KB	-
<input type="checkbox"/> notificaciones	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	10	InnoDB	utf8_general_ci	32 KB	-
<input type="checkbox"/> presentacion	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	64 KB	-
<input type="checkbox"/> profesor	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	48 KB	-
<input type="checkbox"/> proyecto	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	32 KB	-
<input type="checkbox"/> solicitud	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	16 KB	-
<input type="checkbox"/> tribunal	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	48 KB	-
<input type="checkbox"/> usuario	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8_general_ci	32 KB	-
<input type="checkbox"/> valoracion	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	64 KB	-
14 tablas	Número de filas	25	InnoDB	utf8_general_ci	576 KB	0 B

Figura 29. Vista de la base de datos importada.

El siguiente paso a dar es la conexión con nuestra aplicación en PHP. Esta conexión se hará con un patrón singleton, es decir, instanciando la conexión con la Base de Datos para ahorrar recursos. Para la conexión, necesitaremos crear un usuario en MySQL para no emplear el root de MySQL por defecto y poder, asimismo, asignar los privilegios necesarios.

Información de la cuenta

Nombre de usuario:

Use el campo de texto

Admin

Nombre de Host:

Cualquier servidor

%

Contraseña:

Use el campo de texto

Debe volver a escribir:

Complemento de autenticación

Native MySQL authentication

Generar contraseña:

Generar

Base de datos para la cuenta de usuario

☐ Crear base de datos con el mismo nombre y otorgar todos los privilegios.

☐ Otorgar todos los privilegios al nombre que contiene comodín (username_%).

Privilegios globales

☐ Seleccionar todo

Figura 30. Nueva cuenta de administrador para la base de datos.

Y en privilegios seleccionamos los correspondientes a nuestra Base de Datos.

Una vez tenemos nuestra Base de Datos y nuestro usuario, configuramos nuestro archivo PHP de la siguiente manera:

```
<?php
class DB {

    private static $instance;

    private function __construct(){

    }

    public static function getInstance(){
        if(!self::$instance){

            $conex = new mysqli('host','usuario','contraseña','nombre_base_de_datos');

            if (mysqli_connect_errno()) {
                printf("Connect failed: %s\n", mysqli_connect_error());
                exit();
            }
            self::$instance = $conex;
        }
        return self::$instance;
    }

}
```

Figura 31. Conexión a la base de datos mediante PHP.

Mediante esta conexión podremos realizar cualquier intercambio de información entre la Base de Datos y nuestros Modelos. Los siguientes comandos son un ejemplo de una petición a la Base de Datos:

```
/* Carga las notificaciones */
public function getNotificaciones(){

    $db = DB::getInstance();
    $query = "SELECT * FROM NOTIFICACIONES WHERE (DESTINADO = '$this->dni')";
    $stmt = $db->prepare($query);
    $stmt->execute();
    $res = $stmt->get_result();

    $notificaciones = [];
    while($row = $res->fetch_array(MYSQLI_ASSOC)){
        $notificacionAux = new Notificacion($row['ID'],$row['BADGE'],$row['DESTINADO'],$row['MENSAJE'],$row['RUTA']);
        array_push($notificaciones, $notificacionAux);
    }
    return $notificaciones;
}
```

Figura 32. Ejemplo de petición a la base de datos mediante PHP.

Importante añadir siempre la clase de conexión para poder usar nuestra instancia.

```
include_once $_SERVER['DOCUMENT_ROOT'] . "/tfg/conexDB.php";
```

Figura 33. Inclusión de archivos necesarios para la conexión a la base de datos en PHP.

Nota: Se han presentado ciertos problemas al subir la aplicación a un servidor web, ya que este no tenía la misma versión de MySQL. Por tanto, hubo que realizar ciertos cambios a la hora de obtener los resultados de una petición a la base de datos, quedando de la siguiente manera:

```
$stmt->execute();
$res = DB::get_result($stmt);

$notificaciones = [];
while($row = array_shift($res)){
    $notificacionAux = new Notificacion(
        $row['id'], $row['descripcion'], $row['fecha']
    );
}
```

Figura 34. Cambios en la obtención de datos.

Esto es debido a que ciertos servidores, si no tenemos acceso a la instalación de módulos propios, no poseen el driver MySQL Native Driver. Por tanto, PHP no tendría acceso a ciertas funciones como `get_result` o `fetch_array`. En consecuencia, hemos creado una nueva función `get_result` de la siguiente manera:

```
public static function get_result( $Statement ) {
    $RESULT = array();
    $Statement->store_result();
    for ( $i = 0; $i < $Statement->num_rows; $i++ ) {
        $Metadata = $Statement->result_metadata();
        $PARAMS = array();
        while ( $Field = $Metadata->fetch_field() ) {
            $PARAMS[] = &$RESULT[ $i ][ $Field->name ];
        }
        call_user_func_array( array( $Statement, 'bind_result' ), $PARAMS );
        $Statement->fetch();
    }
    return $RESULT;
}
```

Figura 35. Función para la obtención de datos.

Esta es una alternativa posible, si nuestro servidor web donde alojemos la aplicación no posee dicho driver.

Detalles de Estructuración

En el siguiente apartado se describirá cada uno de los archivos pertenecientes a la web, para una mayor comprensión de la estructuración.

Detalles de una clase

El proyecto en sí, consta de varias clases, encargadas de ser nuestro Modelo en el patrón MVC, y las que se encargan de recuperar y guardar los datos en nuestra Base de Datos.

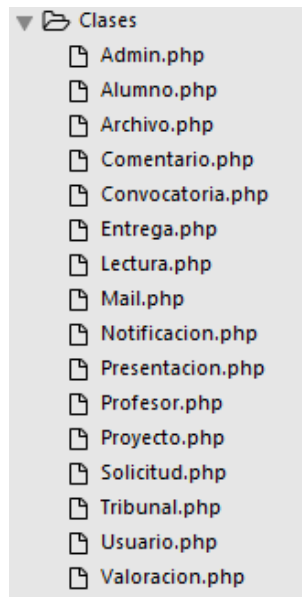


Figura 36. Clases de la aplicación en PHP.

Cada una de estas clases presenta una misma estructura. Cada una de ellas tiene una sección de inclusión, donde se añadirán otras clases necesarias para el uso de esta misma (como puede ser la conexión a la base de datos) y la definición de la clase en sí, donde encontramos: primero una sección de definición de variables de clase, un constructor de clase, una sección de métodos set y get de clase, una sección de métodos de la clase, una sección de métodos estáticos de la clase, y por ultimo una sección de CRUD de clase.

```
<?php
//Clase Ejemplo
include_once $_SERVER['DOCUMENT_ROOT'] . "/tfg/conexDB.php";

class Ejemplo {
    /* Variables */
    private $variable1;
    /* Constructor */
    public function __construct(){
    }

    /* METODOS GET */
    public function getVariable1(){
        return $variable1;
    }
    /* METODOS DE CLASE */
    public function funcionEjemplo(){
    }
    /* METODOS STATIC */
    public static function funcionStaticEjemplo(){
    }
    /* CRUD */

    /* Crear */
    public static function crear(){
    }

    /* Obtener */
    public static function get(){
    }

    /* Actualizar */
    public static function actualizar(){
    }

    /* borrar */
    public static function borrar(){
    }

    /* Funciones auxiliares para el CRUD */
}
?>
```

Figura 37. Estructura de una clase en PHP.

Detalles de un controlador

Cada sección de la plataforma consta de dos tipos de controladores: uno es el controlador de sección, es decir, el encargado de gestionar el CRUD de esa sección, y el otro es el FrontController, que se encuentra en cada vista de la web, y es el que se encarga de recuperar los datos que esa vista necesita, y el control de acceso a ella.

En un controlador de sección se incluyen las clases necesarias para esa sección, una comprobación de inicio de sesión, una comprobación de rol del usuario, y un selector de acción, que será el encargado de realizar dicha acción.

```
<?php
|
session_start();
include_once $_SERVER['DOCUMENT_ROOT'] . "/tfg/Clases/Usuario.php";

if(!empty($_SESSION['usuario'])){
    $usu = new Usuario($_SESSION['usuario']);
    if($usu->getRol()=='rolNecesario'){
        switch ($_POST['action']) {
            case 'action':
                break;
            case 'action2':
                break;
            default:
                echo "Ninguna accion especificada";
                break;
        }
    }else{
        echo "No es rol correcto";
    }
}else{
    echo "No ha iniciado sesion";
}
?>
```

Figura 38. Estructura de un controlador en PHP.

En el FrontController se incluyen las clases necesarias y comprobaciones de inicio de sesión, una de que el usuario posee el rol correcto, y de acceso al recurso, donde se indicara si ese usuario tiene permisos de acceso. Si todo lo demás es correcto, se recuperan los recursos necesarios para mostrar en la vista.

```
<?php
session_start();

$server = $_SERVER['DOCUMENT_ROOT'] . "/tfg/";
$server_name = 'http://'. $_SERVER['SERVER_NAME'] . '/tfg/';

include_once $server . "Clases/Usuario.php";
include_once $server . "Clases/Recurso.php";

if(!empty($_SESSION['usuario'])){
    $usuario = new Usuario($_SESSION['usuario']);
    if($usuario->getRol()=='rolNecesario'){
        if($usuario->accesoA($recurso)){
            $recurso = Recurso::get($idRecurso);
        }
    }
}

<!-- Vista de la pagina -->
```

Figura 39. Estructura de un FrontController en PHP.

Detalles de una vista

Por último, mostraremos la estructura de una vista, que se hará cargo de la interacción entre el usuario y el sistema.

Todas las vistas tendrán la siguiente estructura: primero encontramos las etiquetas donde se especifica el lenguaje que vamos a usar, en nuestro caso español, y con la codificación UTF-8; después encontramos el título de la página, las referencias tanto a los archivos css como los archivos js, las etiquetas meta (que se harán cargo de identificar el dispositivo con el que estamos viendo la web y darle la escala correcta a la vista), y el cuerpo de la web, donde especificaremos el diseño de la vista, añadiendo la cabecera y la sección de la vista.

```
<!-- Vista de la pagina -->

<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8">
    <title>Titulo</title>
    <link rel="stylesheet" type="text/css" href="...">
    ...
    <script type="text/javascript" src="..."></script>
    ...
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
  </head>

  <body>
    <header>
    </header>

    <section>
    </section>

    <footer>
    </footer>
  </body>
</html>

<!-- Fin de FrontCtrl -->
```

Figura 40. Estructura de una vista en HTML.

Implementación de una funcionalidad

Una vez explicado cómo se encuentra estructurada la aplicación web, se va a especificar cómo es la interacción entre un usuario y el sistema, dando de ejemplo sobre cómo un usuario administrador modifica los datos de un proyecto.

En primer lugar, el administrador accede al proyecto que desea modificar.

```
<a href="detallarProyecto.php?id=<?=$idproyecto;?>" class="cajaEntregas">
  <h4 class="list-group-item elemEntregas">
    <?=$nombre;?>
    <span class="badge">Palabras clave: <?=$pc;?></span>
  </h4>
</a>
```

Figura 41. Link a una vista indicando un parámetro mediante GET.

El sistema valida el inicio de sesión del administrador, comprueba su rol y comprueba el acceso a ese recurso (FrontCtrl).

```
<?php
//Comprobamos que la sesion iniciada es la correcta, y si lo es cargamos la pagina
session_start();

$server = $_SERVER['DOCUMENT_ROOT'] . "/tfg/";
$server_name = 'http://'. $_SERVER['SERVER_NAME'] . '/tfg/';

include_once $server . "Clases/Admin.php";
include_once $server . "Clases/Proyecto.php";
include_once $server . "Clases/Valoracion.php";

if(!empty($_SESSION['usuario']) && !empty($_GET['id'])){
    $admin = new Admin($_SESSION['usuario']);
    if($admin->getRol()=='Admin'){
        $proyecto = Proyecto::get($_GET['id']);
        $idProyecto = $proyecto->getID();
    }
}
?>
```

Figura 42. Validación del FrontController.

Si todo esto es correcto recupera el recurso específico y lo muestra al administrador.

```
<section>

<div class="container containerEntregas">
    <div id="idProyecto" value="<?=$idProyecto;?>"></div>

    <div class="row">
        <label class="col-md-2 col-md-offset-2 col-xs-3 control-label">Proyecto</label>
        <div class="col-md-6 col-xs-6">
            <p><?=$proyecto->getNombre(); ?></p>
        </div>
    </div>

    <div class="row">
        <label class="col-md-2 col-md-offset-2 col-xs-3 control-label">Tutor/es asignado/s</label>
        <div class="col-md-6 col-xs-6">
            <p><?=$proyecto->getLinkTutor(); ?></p>
        </div>
    </div>

    <div class="row">
        <label class="col-md-2 col-md-offset-2 col-xs-3 control-label">Alumno/s asignado/s</label>
        <div class="col-md-6 col-xs-6">
            <p><?=$proyecto->getLinkAlumno(); ?></p>
        </div>
    </div>

</div>
```

Figura 43. Vista para detallar un proyecto.

Para recuperar dicho recurso, se accede al modelo que representa a dicho recurso, es decir una Clase (Modelo). Esta Clase recupera una instancia de la base de datos, y mediante sus métodos recupera el recurso desde la base de datos y lo devuelve como un Objeto, el cual es recuperado por el FrontCtrl y mostrado en la vista al administrador.

```
/* Obtener */
public static function get($id){
    $db = DB::getInstance();
    $query = "SELECT * FROM PROYECTO WHERE (ID = ?) LIMIT 1";
    $stmt = $db->prepare($query);
    $stmt->bind_param('i',$id);
    if($stmt->execute()){
        $res = $stmt->get_result();
        $row = $res->fetch_array(MYSQLI_ASSOC);
        $proyecto = new Proyecto($row['ID'],$row['NOMBRE'],$row['DESCRIPCION'],$row['NOTA_FINAL'],$row['PALABRAS_CLAVE'],$row['ESTADO'],$row['PETICION_CONVOCATORIA']);
        return $proyecto;
    }
    return false;
}
```

Figura 44. Obtención de un proyecto desde la base de datos.

Una vez que el administrador tiene la vista cargada, puede decidir qué acción realizar interactuando con la vista. En este caso, el administrador añadirá un usuario nuevo a un proyecto.

Por tanto, el administrador hará clic en el botón modificar proyecto, y se abrirá un modal. En el modal introducirá el nombre del usuario y hará clic en el botón de buscar. Por tanto, el sistema hará una petición mediante Ajax al controlador de Proyectos para buscar ese usuario y mostrarlo al administrador.

```
//Busqueda de usuarios
$('#buscarProfe').on('click',function(){
    busqueda = $('#busquedaProfe').val();
    Buscar(busqueda,'Profesor');
});
$('#buscarAlumno').on('click',function(){
    busqueda = $('#busquedaAlumno').val();
    Buscar(busqueda,'Alumno');
});

function Buscar(busqueda,rol){

    $('#lista'+rol).empty();

    if(busqueda.length > 0){

        var usuario = new Object({
            nombre: busqueda,
            idProyecto: $('#idProyecto').attr('value'),
            rol: rol,
            action: 'buscarUsuario'
        });

        $.post("ctrlProyectos.php",usuario)
        .success(function(resp,estado,jqXHR){

            resp = JSON.parse(resp);

            if(resp['estado']=='ok'){
                //Hay busqueda
                rellenarUsuarios(resp['usuarios'],$('#lista'+rol),'seleccionar'+rol);
            } else {
                //Muestro que no hay resultados
                $('#<div>')
                .html('No hay resultados para esa busqueda')
                .appendTo($('#lista'+rol));
            }
        })
        .error(function(){
            console.log("Error en la peticion");
        });
    }
}
```

Figura 45. Petición mediante AJAX.

Una vez se ha realizado la petición mediante Ajax al controlador de Proyectos.

```
case 'buscarUsuario':

    if(!empty($_POST['nombre']) && !empty($_POST['idProyecto']) && !empty($_POST['rol'])){

        $usuarios = Usuario::getUsuarios($_POST['nombre'],$_POST['idProyecto'],$_POST['rol']);

        if($usuarios!=null){
            $arr = array(
                'estado'=>'ok',
                'usuarios'=>$usuarios
            );
            echo json_encode($arr);
        }else{
            $arr = array(
                'estado'=>'not',
                'usuarios'=> NULL
            );
            echo json_encode($arr);
        }
    }
    break;
```

Figura 46. Interacción del controlador con el modelo.

El controlador obtiene la acción y los recursos necesarios para obtenerla información que necesita, que en este caso es el nombre del usuario y el id del proyecto.

El Sistema realiza la búsqueda mediante la clase Usuario y el siguiente método.

```
/* Devuelve posibles usuarios por rol para el proyecto dado */
public static function getUsers($nombre,$idProyecto,$rol){
    $db = DB::getInstance();
    $nombre = '%' . $nombre . '%';
    $query = "SELECT DNI FROM USUARIO WHERE CONCAT_WS(' ',NOMBRE,APELLIDOS) LIKE ('$nombre') AND ROL = '$rol'";
    if($rol=='Alumno'){
        $query = $query . "AND DNI NOT IN (SELECT DNI FROM ALUMNO)";
    }else{
        $query = $query . "AND DNI NOT IN (SELECT DNI FROM PROFESOR WHERE PROYECTO = '$idProyecto')";
    }

    $stmt = $db->prepare($query);
    $stmt->execute();
    $res = $stmt->get_result();

    //Rellenamos un array con los dni de todos los posibles candidatos a ser tribunal
    $usuarios = [];
    while($row = $res->fetch_array(MYSQLI_ASSOC)){
        $usuario = new Usuario($row['DNI']);
        array_push($usuarios, $usuario->toJSON());
    }
    return $usuarios;
}
```

Figura 47. Obtención de usuarios desde la base de datos.

Una vez obtenidos esos usuarios, JQuery se encarga de mostrarlos al administrador.

```
/* Funcion que rellena los usuarios en el panel de admin */
function rellenarUsuarios(lista,container,id){

    $.each(lista,function(key,value){

        $('<li>')
            .addClass('media')
            .attr('value',value['dni'])
    })
}
```

Figura 48. Función en JavaScript que rellena los usuarios de forma dinámica.

Una vez mostrados, el administrador hace clic en guardar, y el Sistema manda la petición mediante Ajax al controlador de Proyectos, para que lo actualice.

```
var proyecto = new Object({
    nombre: $('#nuevoNombre').val(),
    descripcion: $('#nuevaDescripcion').val(),
    idProyecto: $('#idProyecto').attr('value'),
    profesores: profesores,
    alumnos: alumnos,
    nota_final: $('#nuevaNota').val(),
    palabras_clave: $('#nuevasPalabras').val(),
    action: 'actualizarProyecto'
});

$.post("ctrlProyectos.php",proyecto)
    .success(function(resp,estado,jqXHR){

        if(resp){
            console.log(resp);
            window.location.reload();
        } else {
            console.log(resp);
        }

    })
    .error(function(){
        console.log("Error en la petición");
    });
```

Figura 49. Petición mediante AJAX.

Cuando la petición le llega al controlador, obtiene los datos necesarios y ejecuta la acción especificada.

```
case 'actualizarProyecto':
    if(
        !empty($_POST['nombre']) &&
        !empty($_POST['descripcion']) &&
        !empty($_POST['nota_final']) &&
        !empty($_POST['palabras_clave']) &&
        !empty($_POST['idProyecto'])
    ){
        $idProyecto = $_POST['idProyecto'];
        $nombre = $_POST['nombre'];
        $descripcion = $_POST['descripcion'];
        $palabras_clave = $_POST['palabras_clave'];
        $nota_final = $_POST['nota_final'];
        //Creamos los arrays de alumnos y tutores
        $profesores = $_POST['profesores'];
        $alumnos = $_POST['alumnos'];
        //Guardamos el proyecto en la BBDD
        Proyecto::actualizar($idProyecto,$nombre,$descripcion,$palabras_clave,$nota_final,$profesores,$alumnos);
    }
```

Figura 50. Interacción del controlador con el modelo.

Y actualiza la Base de Datos mediante la clase proyecto.

```
/* Actualizar */
public static function actualizar($id,$nombre,$descripcion,$palabras_clave,$nota,$tutor,$alumno){

    $db = DB::getInstance();
    //Guardamos el proyecto
    $query = "UPDATE PROYECTO SET NOMBRE = ?, DESCRIPCION = ?, PALABRAS_CLAVE = ?, NOTA_FINAL = ? WHERE ID = ?";
    $stmt = $db->prepare($query);
    $stmt->bind_param('ssssi',$nombre,$descripcion,$palabras_clave,$nota,$id);
    $stmt->execute();

    Profesor::borrarAll($id);
    Alumno::borrarAll($id);

    //Guardamos los profesores
    for ($i=0; $i < sizeof($tutor); $i++) {
        Profesor::crear($tutor[$i],$id);
    }

    //Guardamos los alumnos asignados
    for ($i=0; $i < sizeof($alumno); $i++) {
        Alumno::crear($alumno[$i],$id);
    }
    return true;
}
```


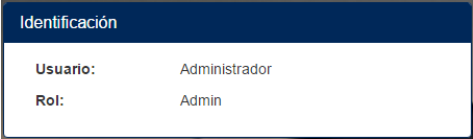
Figura 51. Actualizar datos en la base de datos.


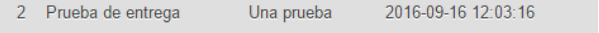
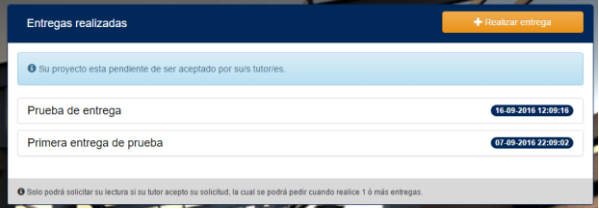
La vista de detalle del proyecto es avisada de los cambios y mediante JQuery se recarga mostrando los nuevos datos.

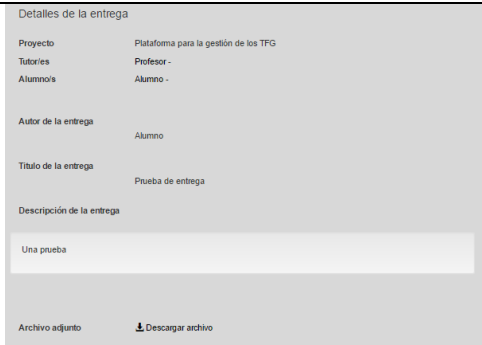

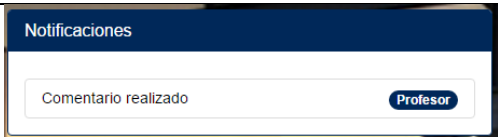
7. Casos de Prueba

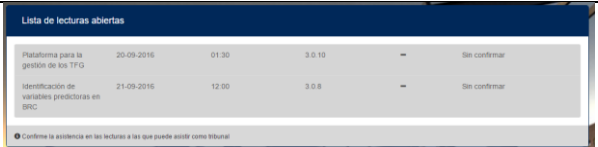
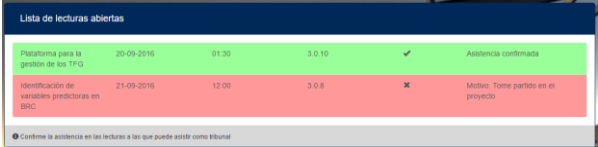
Para los casos de prueba, hemos usado varios usuarios para probar todas las funcionalidades de la web desde los diferentes roles.

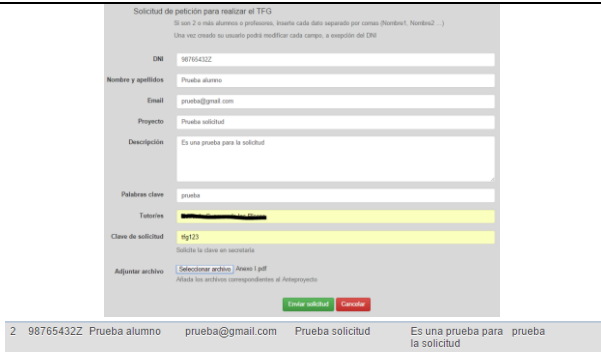
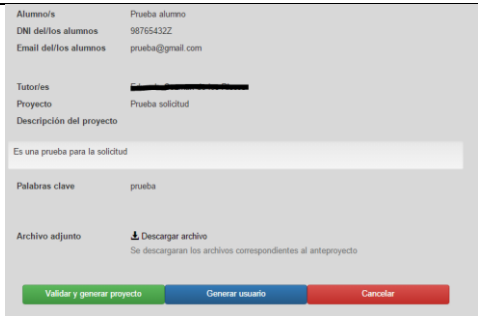
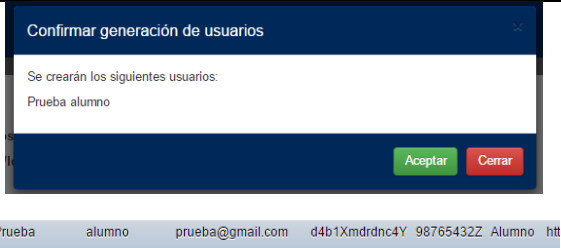
Mostraremos varios casos de prueba como ejemplo de varias funcionalidades de la web.

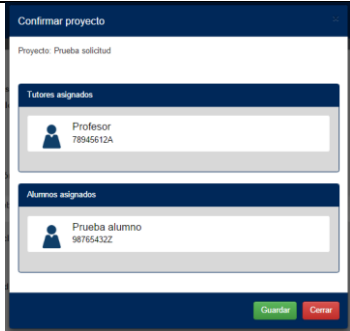
Título	CP01. Acceso a la web	
Propósito	Comprobar que el sistema valide las credenciales de un usuario y lo redirija a su panel principal	
Prerrequisitos	Tener un usuario en el sistema	
Autor	Juan José Ariza Fernández	
Paso	Resultado esperado	Resultado real
1. Insertar datos del usuario y hacer clic en entrar		 <p><i>Figura 52. Inicio de sesión.</i></p>
2. Hacer clic en entrar	Redirección a página principal según su rol	 <p><i>Figura 53. Panel de identificación.</i></p>


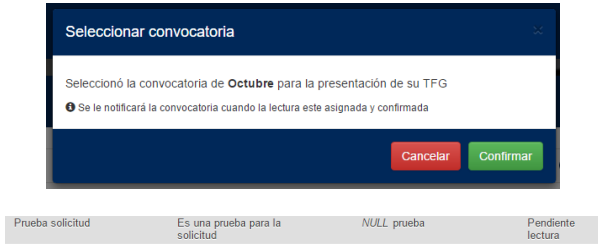

Título	CP02. Realizar entrega	
Propósito	Comprobar que, si un alumno realiza una nueva entrega, ésta se guarde correctamente.	
Prerrequisitos	Tener un proyecto asignado y ser alumno	
Autor	Juan José Ariza Fernández	
Paso	Resultado esperado	Resultado real
1. Hacer clic en realizar entrega	Redirección a sección de realizar entrega	 <p><i>Figura 54. Rellenar entrega.</i></p>
2. El alumno rellena los datos y hace clic en subir entrega (título de la entrega: Prueba de entrega)	EL sistema guarda la entrega en la base de datos	 <p><i>Figura 55. Entrega en la base de datos.</i></p>
3. El alumno ve su entrega en el panel de entregas realizadas	El sistema al guardar la entrega redirige al alumno al panel principal	 <p><i>Figura 56. Panel de entregas.</i></p>


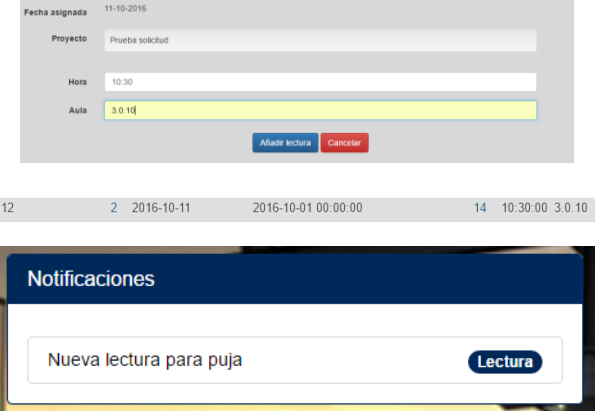
Título	CP03. Corrección de una entrega	
Propósito	Comprobar que el tutor de un alumno que haya realizado una entrega sea visible por su tutor y pueda corregirla o agregar un comentario.	
Prerrequisitos	Tener un proyecto asignado y ser tutor	
Autor	Juan José Ariza Fernández	
Paso	Resultado esperado	Resultado real
1. Hacer clic en notificación de entrega realizada o en la entrega dentro del proyecto	Redirección a sección de detalles de la entrega	 <p><i>Figura 57. Detalles de una entrega.</i></p>
2. El profesor revisa la entrega y realiza un comentario	EL sistema guarda el comentario realizado en la base de datos	 <p><i>Figura 58. Comentario realizado.</i></p>
	El sistema notifica al alumno	 <p><i>Figura 59. Panel de notificaciones.</i></p>

Título	CP04. Pujar en una lectura	
Propósito	Comprobar que un profesor pueda pujar en una lectura abierta y se guarde su respuesta para que el administrador tenga acceso a posibles tribunales para una lectura	
Prerrequisitos	Ser profesor y haber lecturas abiertas	
Autor	Juan José Ariza Fernández	
Paso	Resultado esperado	Resultado real
1. Hacer clic en botón pujar	El sistema recupera las lecturas abiertas y las muestra al profesor	 <p>Figura 60. Panel de pujas.</p>
2. El profesor hace clic en una lectura e introduce su respuesta	El sistema muestra un modal para seleccionar respuesta	 <p>Figura 61. Respuesta a una puja.</p>
3. El profesor elige su respuesta	El sistema guarda la respuesta	 <p>Figura 62. Panel de pujas actualizado.</p>
4. El administrador accede a una lectura y puede asignar un tribunal		 <p>Figura 63. Panel detalle de una lectura.</p>

Título	CP05. Aceptar una solicitud de TFG	
Propósito	Comprobar que el sistema guarda correctamente una solicitud para ser procesada por la administración	
Prerrequisitos		
Autor	Juan José Ariza Fernández	
Paso	Resultado esperado	Resultado real
1. El alumno rellena una solicitud de TFG	El sistema guarda la solicitud	 <p>Figura 64. Solicitud de TFG.</p>
2. El administrador accede a la solicitud	El sistema recupera la solicitud y muestra la vista de detalles	 <p>Figura 65. Detalle de solicitud de TFG.</p>
3. El administrador hace clic en generar usuarios y confirma	El sistema muestra un modal donde se le indica el usuario nuevo que va a generar, y lo genera	 <p>Figura 66. Generación de usuarios.</p>

4. El administrador hace clic en validar y generar proyecto, y en guardar	El sistema muestra un modal donde se especifican los detalles del proyecto que va a generar y lo genera	 <p>Prueba solicitud Es una prueba para la solicitud NULL prueba</p> <p><i>Figura 67. Generación del proyecto.</i></p>
---	---	--

Título	CP06. Petición de convocatoria	
Propósito	Comprobar que el sistema guarda correctamente la petición de un alumno	
Prerrequisitos	El tutor del alumno debe haber aceptado la petición de autorización del alumno	
Autor	Juan José Ariza Fernández	
Paso	Resultado esperado	Resultado real
1. El alumno hace clic en seleccionar convocatoria	El sistema muestra las convocatorias abiertas	 <p><i>Figura 68. Panel para solicitar convocatoria.</i></p>
2. El alumno elige la que más le conviene	El sistema muestra un modal de confirmación y guarda la selección. La administración es avisada.	 <p><i>Figura 69. Panel de confirmación.</i></p>
3. El administrador hace clic en la notificación para seleccionar una lectura al proyecto	El sistema carga los datos necesarios para asignar la lectura	 <p><i>Figura 70. Panel de notificaciones.</i></p>

<p>4. El administrador hace clic en el día que quiera para asignar la lectura</p>	<p>El sistema carga la vista para detallar la lectura</p>	 <p><i>Figura 71. Panel de asignación de lecturas.</i></p>
<p>5. El administrador rellena los datos y guarda la lectura</p>	<p>El sistema guarda la lectura, y avisa a los profesores de que existe una nueva lectura para pujar</p>	 <p><i>Figura 72. Panel para rellenar nueva lectura.</i></p>

9. Conclusiones y Trabajo Futuro

Actualmente, en la Universidad de Málaga no se dispone de un sistema con el que gestionar la línea de entrega de un Trabajo de Fin de Grado (TFG), por tanto, con este sistema buscamos cubrir todas las necesidades que conllevan la realización del mismo. Ya que, gracias a este sistema, tanto los alumnos, los profesores y el personal administrativo podrán tener un conocimiento más detallado de cada uno de los TFG que se encuentran abiertos y el estado de ellos.

Por otro lado, cabe destacar que la tecnología web es actualmente uno de los medios de comunicación más usados. Por tanto, esta es la manera más efectiva de crear un sistema donde se conecten diferentes usuarios para comunicarse entre sí con un motivo en común. Gracias a esto, cualquier usuario involucrado tendrá acceso desde cualquier punto donde se encuentre.

Para la realización de este sistema se han usado tecnologías que actualmente están siendo muy demandadas, como es PHP, JavaScript, Bootstrap, etc. Gracias a este tipo de tecnologías como es Bootstrap, se ha podido realizar un interfaz adaptable a cualquier dispositivo, ya sea ordenador, tablet o móvil, ya que cada vez más se usan este tipo de dispositivos para acceder a internet.

Con respecto a la implementación de la aplicación, se ha realizado con uno de los patrones de diseño que más se usan actualmente (MVC), ya que se busca que el sistema pueda ser ampliado. Gracias a que el diseño de la aplicación ha seguido este patrón, cualquier desarrollador podría abordar sin mucho esfuerzo la inclusión de nuevas funcionalidades en el sistema. Además de que todas las tecnologías usadas son open-source, y cada una de ellas posee una amplia comunidad que la respalda. Esto facilita enormemente su uso y la resolución de problemas.

Por último, como trabajo futuro se presentan las siguientes ideas, una de las principales tareas es alojar y configurar la aplicación web en un servidor para su puesta en producción y que todo el personal relacionado con la realización de un TFG pueda beneficiarse de las ventajas que aporta esta plataforma, una configuración más detallada de los datos de entrada (validando cada uno de ellos de forma dinámica y avisando al usuario de cualquier error), un sistema anti-plagio para la detección de TFG duplicados, una adaptación completa de la aplicación web con respecto a la gestión de documentos oficiales para la realización de un TFG (entrega anteproyecto, petición de convocatoria, entrega de memorias, etc.) similar a una solicitud de petición de becas para estudios, la implementación de un chat entre los miembros participantes de un TFG independiente a los comentarios de cada entrega, etc. Todo esto puede tomar como base inicial este proyecto, ya que como anteriormente se expuso, gracias a la utilización de este tipo de tecnologías, el desarrollo y ampliación de la plataforma podría ser abordado sin mucho esfuerzo.

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA

Grado en Ingeniería Informática

Anexo I – Manual de Usuario

Append I – User Manual

Realizado por

Juan José Ariza Fernández

Tutorizado por

Eduardo Guzmán de los Riscos

Departamento

Lenguaje y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

MÁLAGA, octubre, 2016

Índice

Usuario	90
Alumno	92
Profesor/Tribunal	96
Administrador	98

Manual de Usuario

En este punto se va a construir una guía de cara a facilitar el uso de la plataforma a los usuarios finales. La guía la estructuraremos en los diferentes roles que encontramos en la plataforma.

Usuario

Portada

Portada de la plataforma web, a la que podrá acceder cualquier usuario que acceda al sistema. En ella tenemos la opción de entrar como un usuario registrado, o realizar una petición de TFG si somos un alumno.

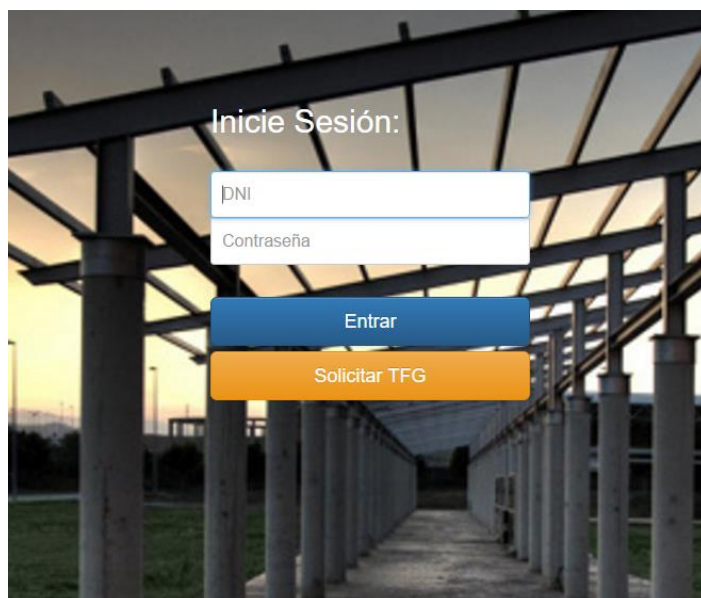


Figura 73. Página principal.

Notificaciones

Panel donde se mostrarán las notificaciones personales de cada usuario.

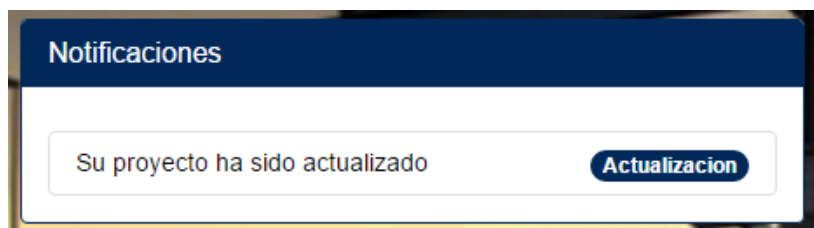
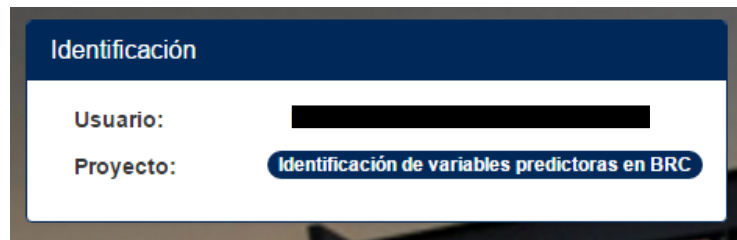


Figura 74. Panel de notificaciones.

Identificación

Panel donde cada usuario será identificado.

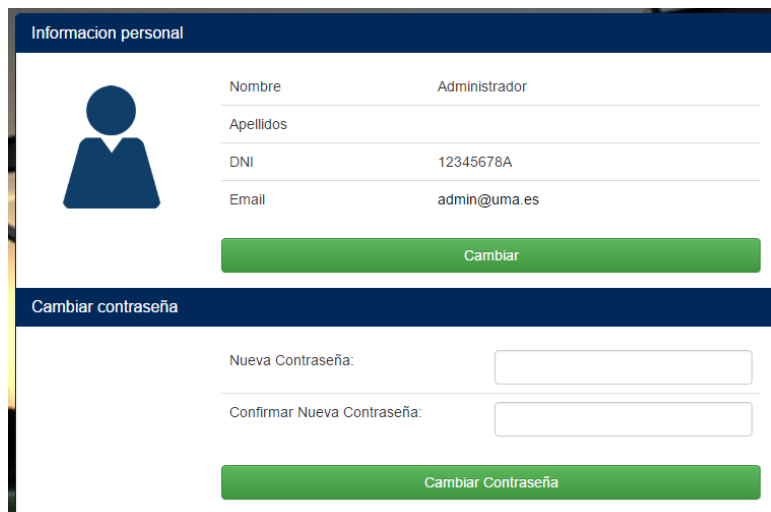
El panel de identificación tiene un encabezado azul oscuro con el título "Identificación" en blanco. Debajo, hay un formulario con dos campos: "Usuario:" con un campo de texto negroado, y "Proyecto:" con un botón azul que dice "Identificación de variables predictoras en BRC".


Identificación	
Usuario:	<input type="text"/>
Proyecto:	<button>Identificación de variables predictoras en BRC</button>

Figura 75. Panel de identificación.

Perfil de Usuario

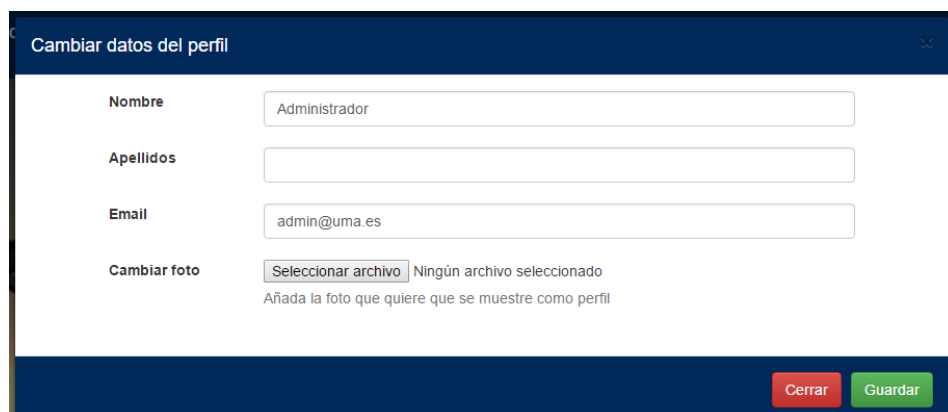
Panel donde cada usuario tendrá acceso a sus datos del perfil, y donde podrá cambiar cualquiera de ellos.

El panel de perfil de usuario está dividido en dos secciones. La primera, "Información personal", muestra un ícono de usuario a la izquierda y una lista de datos a la derecha: Nombre (Administrador), Apellidos, DNI (12345678A) y Email (admin@uma.es). Debajo de estos datos hay un botón verde "Cambiar". La segunda sección, "Cambiar contraseña", tiene dos campos de texto para "Nueva Contraseña:" y "Confirmar Nueva Contraseña:", seguidos de un botón verde "Cambiar Contraseña".

Información personal	
	Nombre: Administrador
	Apellidos:
	DNI: 12345678A
	Email: admin@uma.es
<button>Cambiar</button>	

Cambiar contraseña	
Nueva Contraseña:	<input type="text"/>
Confirmar Nueva Contraseña:	<input type="text"/>
<button>Cambiar Contraseña</button>	

Figura 76. Perfil de usuario.

Este panel se titula "Cambiar datos del perfil". Contiene cuatro campos de entrada: "Nombre" (con el valor "Administrador"), "Apellidos", "Email" (con el valor "admin@uma.es") y "Cambiar foto". El campo "Cambiar foto" incluye un botón "Seleccionar archivo" y el texto "Ningún archivo seleccionado". Debajo de esto, hay una instrucción: "Añada la foto que quiere que se muestre como perfil". En la parte inferior derecha, hay dos botones: "Cerrar" (rojo) y "Guardar" (verde).

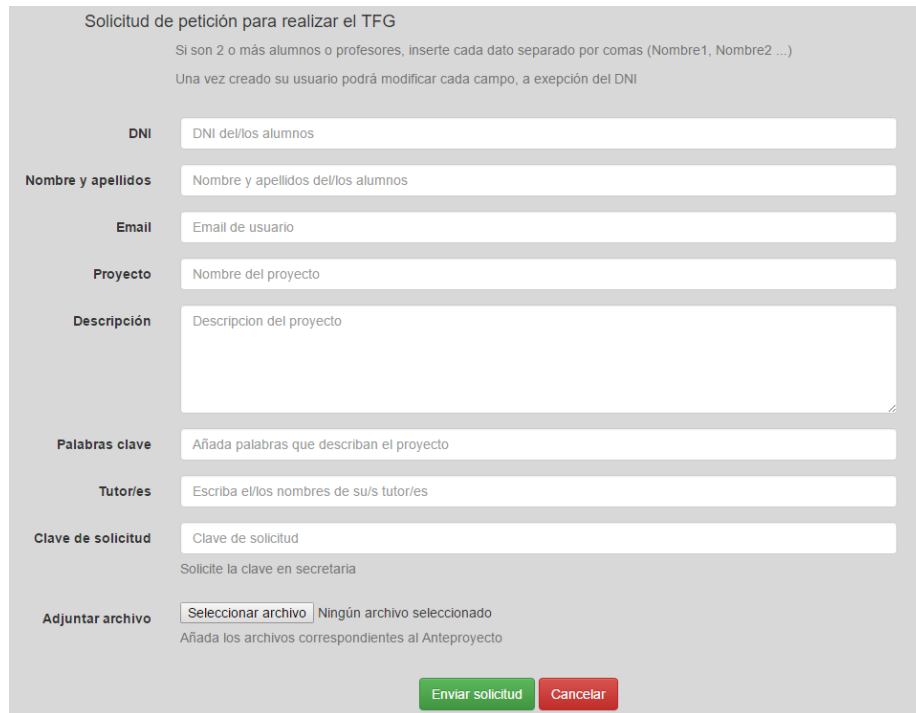
Cambiar datos del perfil	
Nombre	<input type="text" value="Administrador"/>
Apellidos	<input type="text"/>
Email	<input type="text" value="admin@uma.es"/>
Cambiar foto	<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado
Añada la foto que quiere que se muestre como perfil	
<div><button>Cerrar</button><button>Guardar</button></div>	

Figura 77. Modificar perfil de usuario.

Alumno

Solicitar TFG

Panel al que se accede desde la portada de la web, y es el que rellena un alumno cuando desea realizar una petición para hacer su TFG.



Solicitud de petición para realizar el TFG

Si son 2 o más alumnos o profesores, inserte cada dato separado por comas (Nombre1, Nombre2 ...)

Una vez creado su usuario podrá modificar cada campo, a excepción del DNI

DNI

Nombre y apellidos

Email

Proyecto

Descripción

Palabras clave

Tutor/es

Clave de solicitud

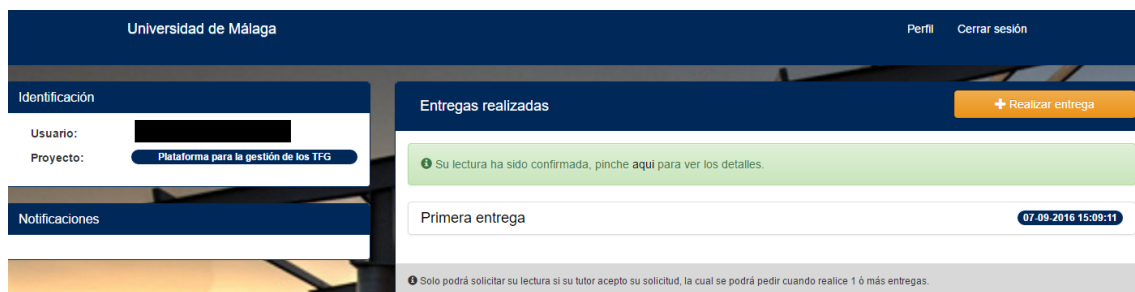
Solicite la clave en secretaría

Adjuntar archivo Ningún archivo seleccionado
Añada los archivos correspondientes al Anteproyecto

Figura 78. Panel de solicitar TFG.

Panel principal

Panel principal donde el alumno podrá ver las entregas que ha realizado, realizar una nueva entrega, y ver el estado en el que se encuentra su lectura, si se ha solicitado.



Universidad de Málaga

Perfil Cerrar sesión

Identificación

Usuario:

Proyecto:

Notificaciones

Entregas realizadas

Su lectura ha sido confirmada, pinche [aquí](#) para ver los detalles.

Primera entrega 07.09.2016 15:09:11

Solo podrá solicitar su lectura si su tutor acepta su solicitud, la cual se podrá pedir cuando realice 1 ó más entregas.

Figura 79. Página principal del alumno.

En el caso de que el alumno aun no haya pedido la autorización a su tutor para pedir una convocatoria, esta será la vista, y en la cual, podrá pedir la autorización a su tutor mediante el botón pedir autorización.

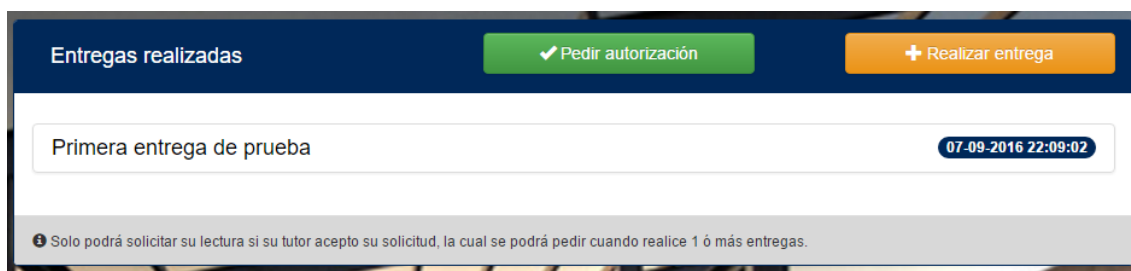


Figura 80. Panel de entregas – Pedir autorización.

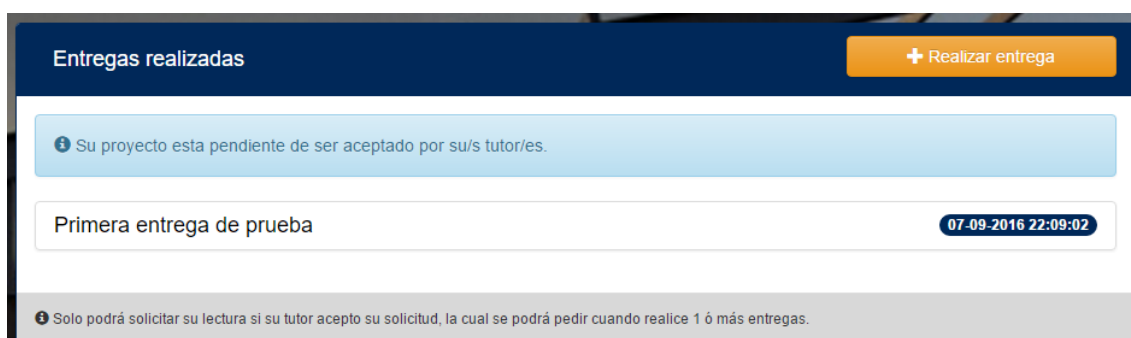


Figura 81. Panel de entregas – A espera de la autorización.

Una vez haya sido autorizado por el tutor, esta será la vista, y con ella podrá seleccionar la convocatoria a la que asistir.

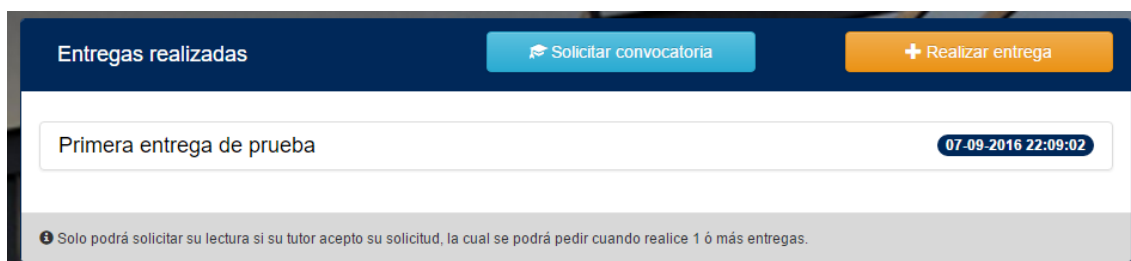


Figura 82. Panel de entregas – Solicitar convocatoria.

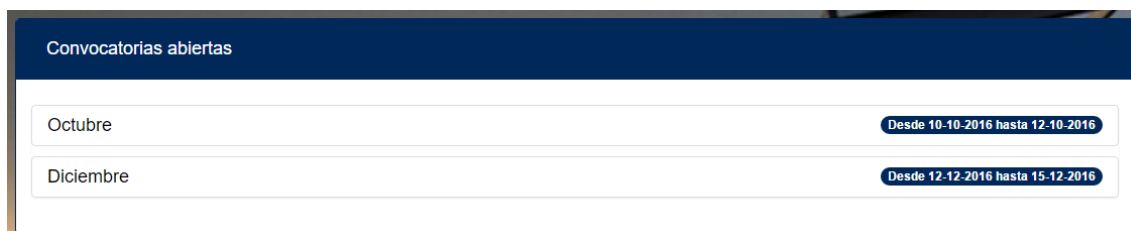


Figura 83. Panel de selección de convocatoria.

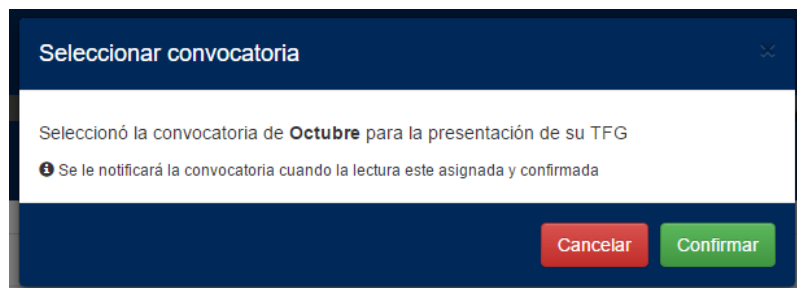


Figura 84. Confirmación de convocatoria.



Figura 85. Aviso pendiente de lectura.

Entregas

El alumno podrá realizar una nueva entrega, o ver una entrega anterior, donde aparecerán los comentarios del profesor, o podrá borrar dicha entrega.

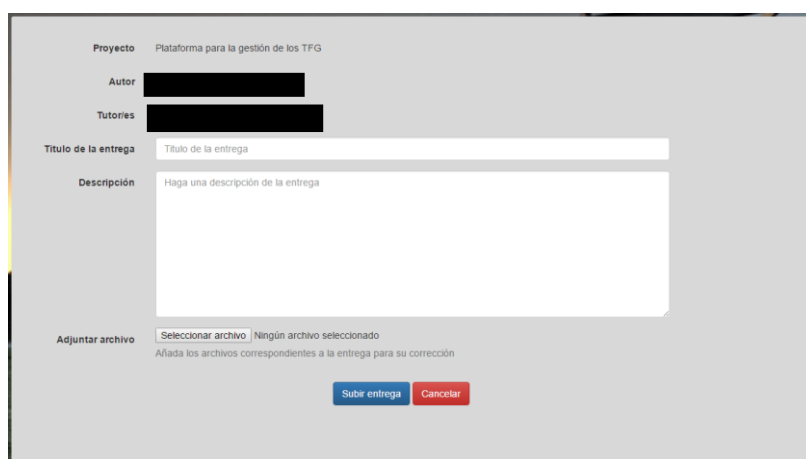


Figura 86. Panel para realizar entregas.

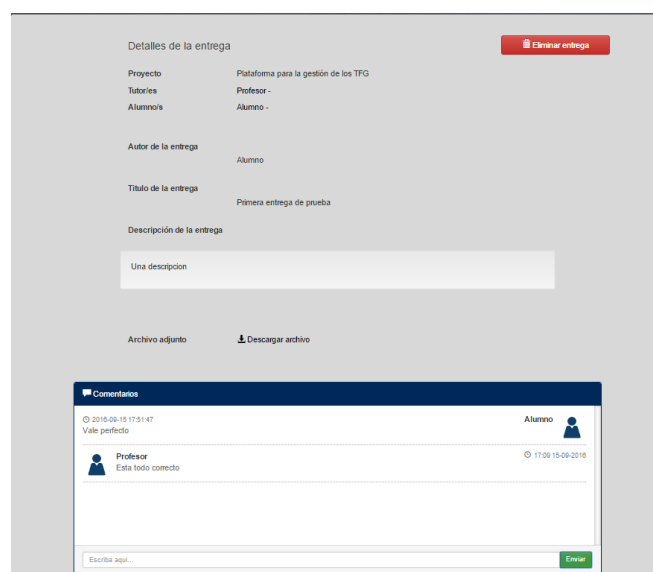
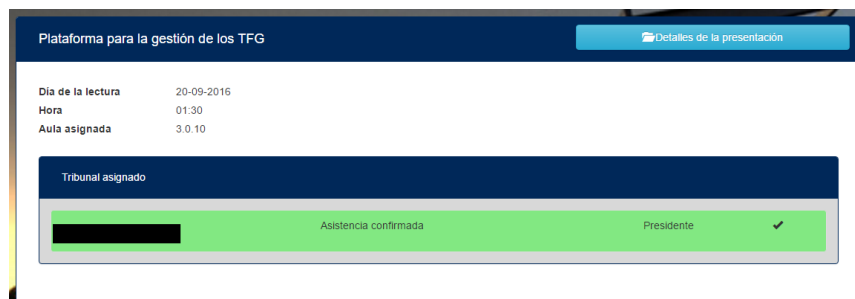


Figura 87. Panel de detalles de entrega.

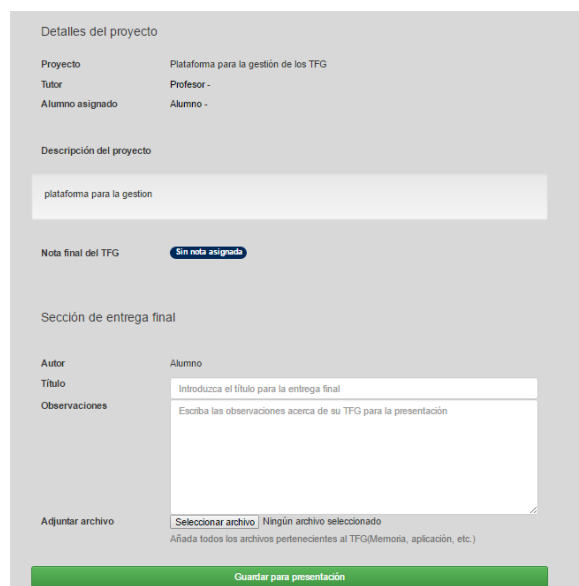
Lectura

Panel donde el alumno tendrá acceso a los detalles de su lectura. Además, tendrá acceso a los detalles de su presentación, donde podrá realizar una entrega final, y ver las valoraciones del tutor y del tribunal.



Plataforma para la gestión de los TFG	
Detalles de la presentación	
Día de la lectura	20-09-2016
Hora	01:30
Aula asignada	3.0.10
Tribunal asignado	
Asistencia confirmada	Presidente

Figura 88. Detalles de la lectura.



Detalles del proyecto

Proyecto: Plataforma para la gestión de los TFG

Tutor: Profesor -

Alumno asignado: Alumno -

Descripción del proyecto

plataforma para la gestion

Nota final del TFG: Sin nota asignada

Sección de entrega final

Autor: Alumno

Título: Introduzca el título para la entrega final

Observaciones: Escriba las observaciones acerca de su TFG para la presentación

Adjuntar archivo: Seleccionar archivo | Ningún archivo seleccionado

Añada todos los archivos pertenecientes al TFG (Memoria, aplicación, etc.)

Guardar para presentación

Figura 89. Detalles de la presentación final.



Valoración del/los tutor/es

Tutor: [Redacted]

Valoración hecha por el Tutor

Muy buen alumno

Nota estimada: 9.00

Valoración del/los tribunal/es

Tribunal: [Redacted]

Valoración hecha por el Tribunal

Gran proyecto si señor

Nota estimada: 9.50

Figura 90. Valoración del tutor y del tribunal.

Profesor/Tribunal

Panel principal

Página donde el profesor podrá ver los proyectos de los que es tutor, además de las entregas que realizan los alumnos, los comentarios y el estado en que se encuentra la lectura del proyecto. Además, tendrá un panel donde se mostrarán las lecturas a las que fue asignado como tribunal.

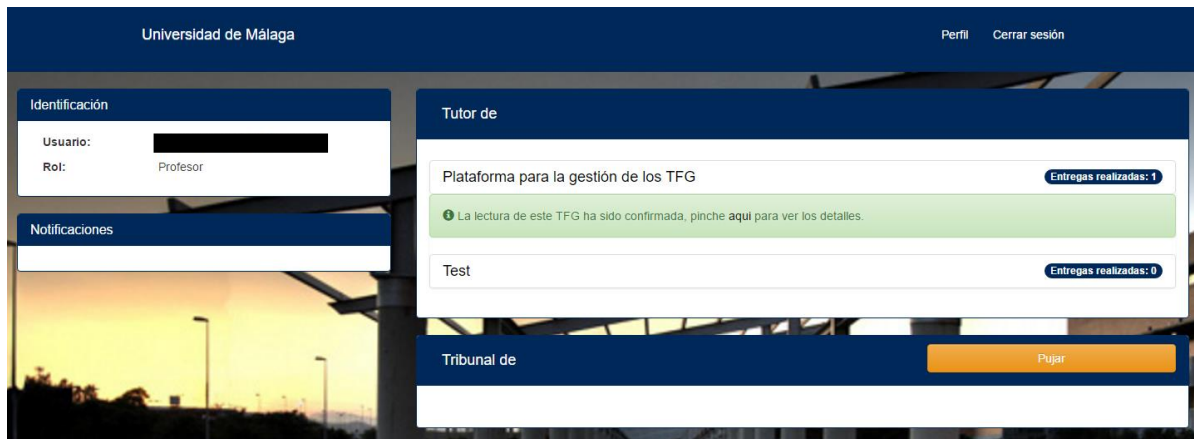


Figura 91. Panel principal del profesor.

Proyectos

Panel donde verá los detalles del proyecto. Tendrá acceso a autorizar la petición de convocatoria del alumno, y a las entregas que realiza cada alumno.

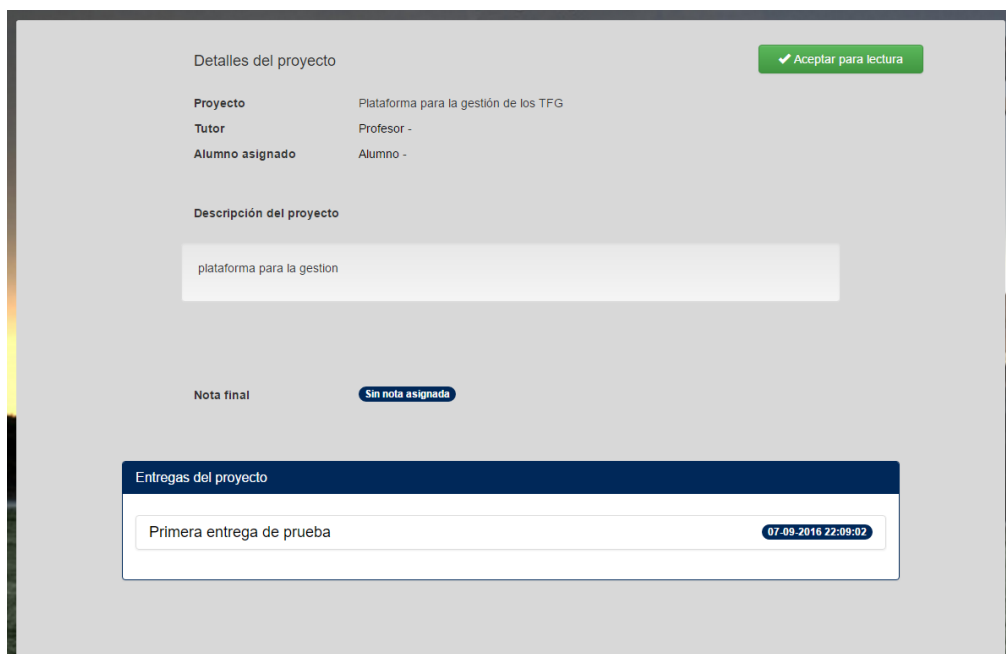


Figura 92. Detalles de un proyecto.

Lecturas

En el panel de pujas, el profesor podrá expresar su interés en ser miembro del tribunal de un TFG. Dicha respuesta, será valorada por el administrador en la asignación del tribunal de un proyecto.



Figura 93. Panel de pujas.

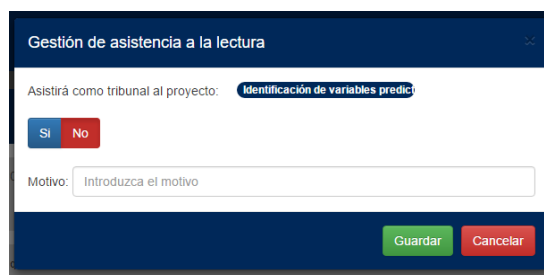


Figura 94. Modal para realizar una puja.

Panel con la lectura asignada.

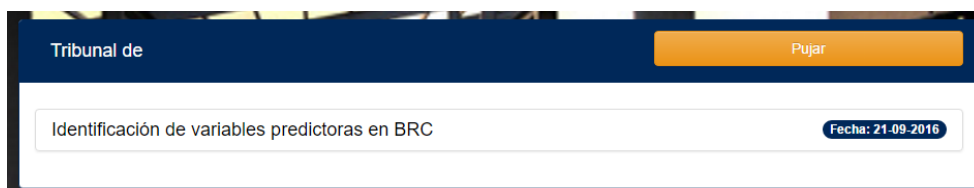


Figura 95. Panel de lecturas asignadas.

Y en el detalle de la lectura, tendrá acceso a la entrega final realizada por el alumno, y a una sección donde calificar al alumno. Añadir que, si es tutor de un proyecto, tendrá el mismo panel donde podrá calificar a su alumno.

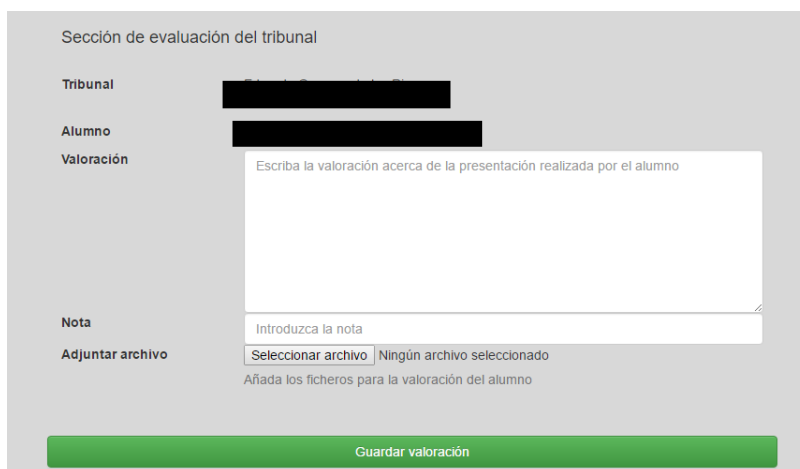


Figura 96. Panel de evaluación del alumno.

Administrador

Panel principal

En esta página el administrador tendrá acceso a la gestión de los recursos de la aplicación, ya sean usuarios, proyectos, solicitudes o convocatorias.



Figura 97. Página principal del administrador.

Solicitudes

Panel donde gestionar las solicitudes de TFG que realizan los alumnos. En cada solicitud el administrador tendrá acceso a generar los usuarios alumnos y a validar y generar el proyecto, para que tanto el alumno como el tutor puedan comunicarse.

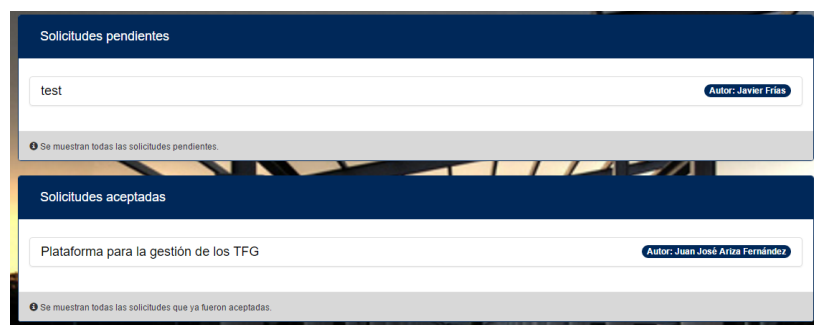


Figura 98. Panel de gestión de las solicitudes.

The screenshot shows the 'Detalles de una solicitud' form. It contains the following fields: 'Alumno/s' (redacted), 'DNI de/los alumnos' (12345678A), 'Email de/los alumnos' (redacted), 'Tutor/es' (Alguno), 'Proyecto' (test), 'Descripción del proyecto' (empty), 'Prueba de anteproyecto' (empty), 'Palabras clave' (tfg), 'Archivo adjunto' (Descargar archivo, with a note 'Se descargarán los archivos correspondientes al anteproyecto'), and three buttons at the bottom: 'Validar y generar proyecto' (green), 'Generar usuario' (blue), and 'Cancelar' (red).

Figura 99. Detalles de una solicitud.

Usuarios

Panel donde el administrador tendrá acceso a la gestión de los usuarios, crear, eliminar, ver, etc.

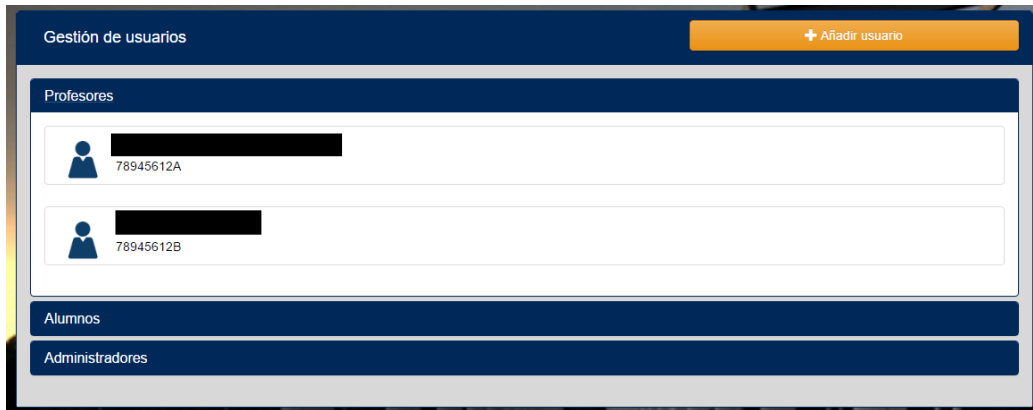


Figura 100. Panel de administración de usuarios.

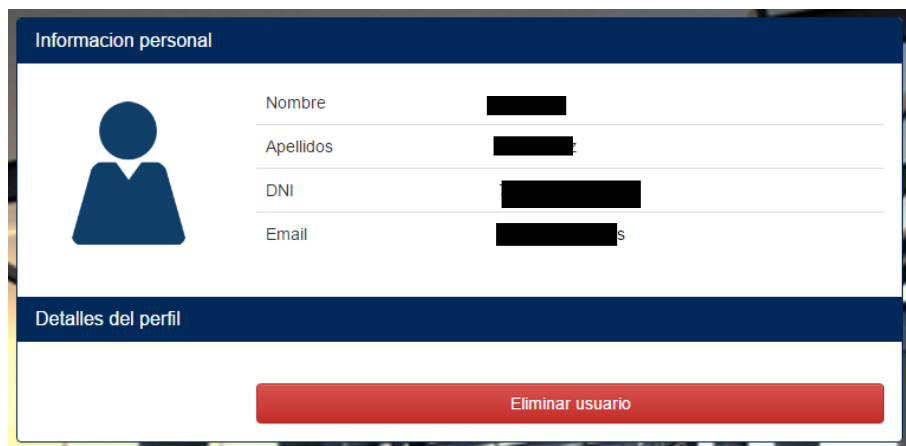


Figura 101. Panel de detalles de un usuario.

Nombre	<input type="text" value="Nombre de usuario"/>
Apellidos	<input type="text" value="Apellidos de usuario"/>
Email	<input type="text" value="Email de usuario"/>
DNI	<input type="text" value="DNI de usuario"/>
Rol	<input type="text" value="Alumno"/>

Aviso: La contraseña del usuario será su DNI

Figura 102. Panel de creación de usuarios.

Proyectos

Panel donde el administrador tendrá acceso a los 10 últimos proyectos abiertos, y a un buscador por categorías. Además de poder crear, ver, modificar o eliminar un proyecto.

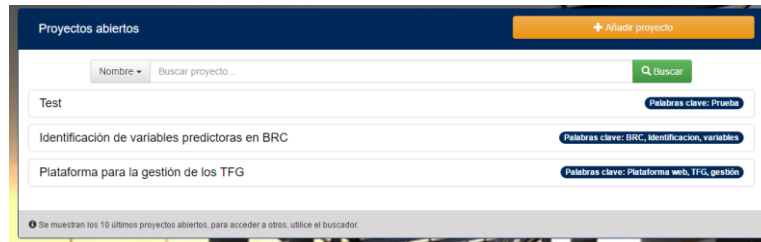


Figura 103. Panel de administración de proyectos.

Este panel muestra los detalles de un proyecto. Al principio, se muestran los campos 'Proyecto' (Plataforma para la gestión de los TFG), 'Tutor/es asignado/s' (Profesor -) y 'Alumno/s asignado/s' (Alumno -). Luego, la 'Descripción del proyecto' se encuentra en un campo de texto con el valor 'plataforma para la gestion'. La sección 'Lectura' indica 'Nota final' como 'Sin nota asignada'. En la parte inferior, hay dos botones: 'Modificar datos' (verde) y 'Eliminar proyecto' (rojo).

Figura 104. Panel de detalles de un proyecto.

En el siguiente panel se muestra el modal que se abre cuando el administrador hace clic en modificar datos.

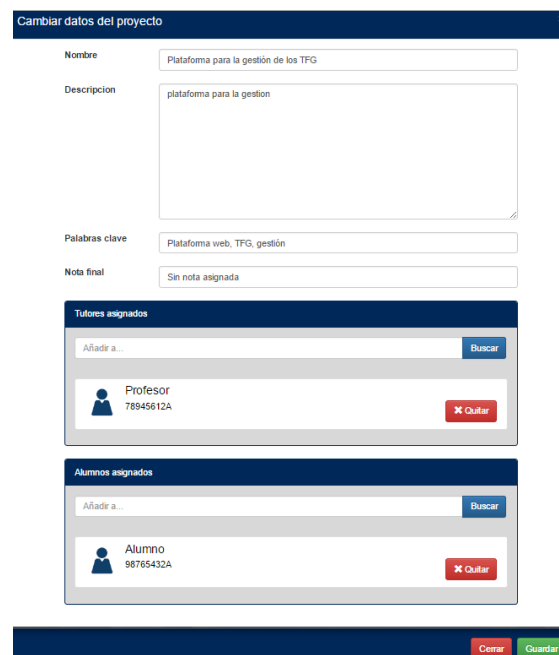
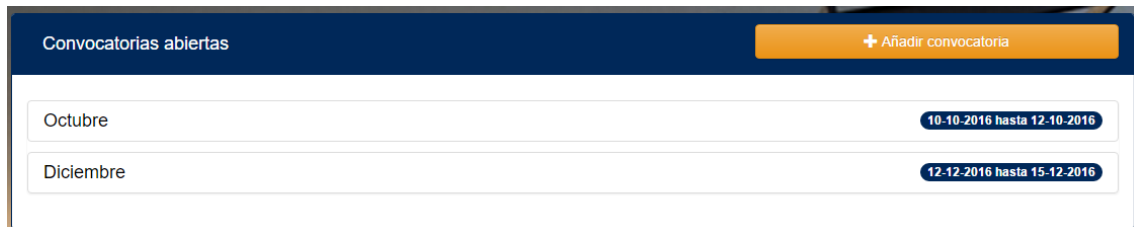
Este modal, titulado 'Cambiar datos del proyecto', permite editar la información de un proyecto. Incluye campos para 'Nombre' (Plataforma para la gestión de los TFG), 'Descripción' (plataforma para la gestion), 'Palabras clave' (Plataforma web, TFG, gestión) y 'Nota final' (Sin nota asignada). Debajo, se muestran las secciones 'Tutores asignados' y 'Alumnos asignados', cada una con un campo 'Añadir a...', un botón 'Buscar' y una lista de usuarios seleccionados (Profesor 78945612A y Alumno 98765432A) con botones 'Quitar'. En la parte inferior del modal, hay botones 'Cerrar' (rojo) y 'Guardar' (verde).

Figura 105. Panel para cambiar los datos de un proyecto.

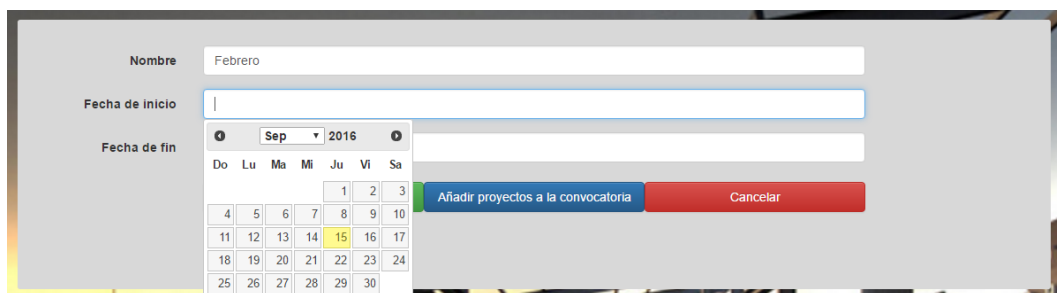
Convocatorias

Panel donde el administrador tendrá acceso a las convocatorias abiertas y a la creación y modificación de ellas. Además de añadir lecturas a cada convocatoria, modificarlas y asignar tribunal a cada una de ellas.



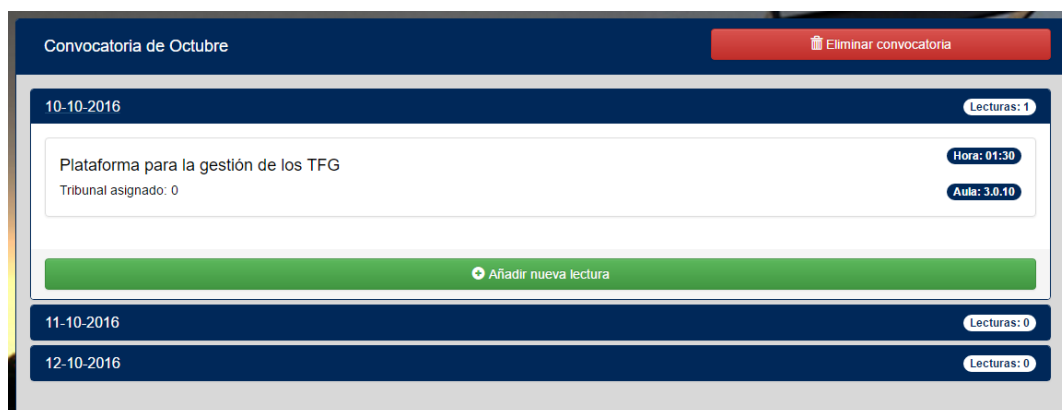
Panel de administración de convocatorias. Muestra un encabezado con el título "Convocatorias abiertas" y un botón "+ Añadir convocatoria". Debajo, hay una lista de convocatorias con los meses "Octubre" y "Diciembre". Cada mes tiene un botón que indica el rango de fechas: "10-10-2016 hasta 12-10-2016" para Octubre y "12-12-2016 hasta 15-12-2016" para Diciembre.

Figura 106. Panel de administración de convocatorias.



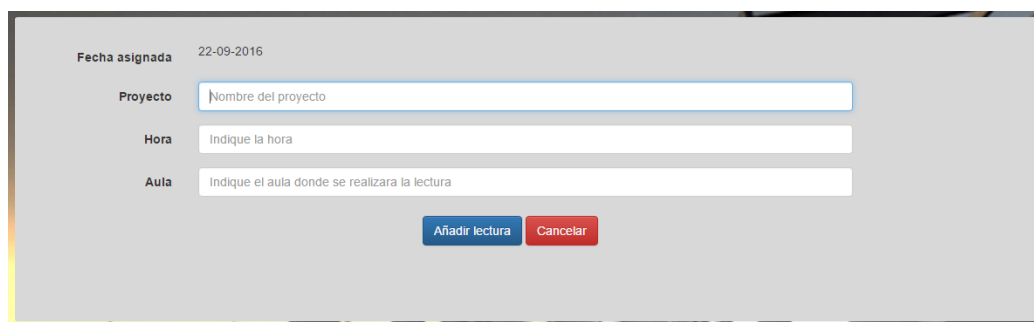
Panel para crear una convocatoria. Incluye campos para "Nombre" (con el valor "Febrero"), "Fecha de inicio" (un campo vacío) y "Fecha de fin" (un calendario para el mes de Septiembre de 2016). Debajo del calendario hay dos botones: "Añadir proyectos a la convocatoria" (azul) y "Cancelar" (rojo).

Figura 107. Panel para crear una convocatoria.



Panel para la gestión de lecturas. Muestra una convocatoria de "Octubre" con un botón "Eliminar convocatoria". Debajo, hay una lista de lecturas para el periodo "10-10-2016". La primera lectura es "Plataforma para la gestión de los TFG" con un tribunal asignado de 0, una hora de 01:30 y un aula de 3.0.10. Hay un botón "Añadir nueva lectura" (verde) y botones "Lecturas: 1", "Lecturas: 0" para las fechas "11-10-2016" y "12-10-2016".

Figura 108. Panel para la gestión de lecturas.



Panel para crear una nueva lectura. Incluye campos para "Fecha asignada" (22-09-2016), "Proyecto" (Nombre del proyecto), "Hora" (Indique la hora) y "Aula" (Indique el aula donde se realizara la lectura). Debajo hay dos botones: "Añadir lectura" (azul) y "Cancelar" (rojo).

Figura 109. Panel para crear una nueva lectura.

Plataforma para la gestión de los TFG
Modificar

Día de la lectura: 10-10-2016
Hora: 01:30
Aula asignada: 3.0.10

Tribunal asignado

Cerrar

Figura 110. Modal de detalles de una lectura.

Plataforma para la gestión de los TFG
Eliminar lectura

Día de la lectura: 20-09-2016
Hora: 01:30
Aula asignada: 3.0.10

Tribunal asignado

Asignado como: Presidente
Quitar

Pulse en quitar si desea eliminar a un tribunal

Posible tribunal

Pulse en añadir para asignar el profesor como tribunal

Figura 111. Panel de detalles de una lectura.

Bibliografía

- Documentación de PHP: <http://php.net/>
- Documentación de JavaScript: <http://www.w3schools.com/jsref/>
- Documentación de JQuery: <https://jquery.com/>
- Documentación de JQueryUI: <https://jqueryui.com/>
- Documentación de CSS: <http://www.w3schools.com/css/>
- Documentación de Bootstrap: <http://getbootstrap.com/>
- Documentación de MySQL: <http://www.mysql.com/>
- Servidor de alojamiento de pruebas: <http://www.hostinger.es/>
 - Servidor de pruebas: <http://localhost/tfg/> (Local) - <http://tfgplatform.esy.es/tfg/> (Red)
- Documentación de Ajax: <http://api.jquery.com/jquery.ajax/>
- Editor de textos: <https://www.sublimetext.com/3>
- XAMPP: <https://www.apachefriends.org/es/index.html>
- Documentación de JSON: <http://www.json.org/>
- Realización de Análisis, Diseño e Implementación:
 - Análisis y Diseño de los Sistemas de Información
 - Asignatura de tercero – Grado en Ingeniería Informática en la Universidad de Málaga (ETSII)
- Realización de la aplicación web:
 - Sistemas de Información para Internet
 - Asignatura de tercero – Grado en Ingeniería Informática en la Universidad de Málaga (ETSII)
- Realización de la base de datos:
 - Bases de Datos
 - Asignatura de segundo – Grado en Ingeniería Informática en la Universidad de Málaga (ETSII)
 - Administración de Bases de Datos
 - Asignatura de tercero – Grado en Ingeniería Informática en la Universidad de Málaga (ETSII)